

Rcpp Version 1.0.9

Generated by Doxygen 1.9.1



---

<b>1 Namespace Index</b>	<b>1</b>
1.1 Namespace List	1
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Class Index</b>	<b>29</b>
3.1 Class List	29
<b>4 File Index</b>	<b>47</b>
4.1 File List	47
<b>5 Namespace Documentation</b>	<b>55</b>
5.1 R Namespace Reference	55
5.1.1 Function Documentation	58
5.1.1.1 <code>bessel_i()</code>	58
5.1.1.2 <code>bessel_i_ex()</code>	58
5.1.1.3 <code>bessel_j()</code>	58
5.1.1.4 <code>bessel_j_ex()</code>	59
5.1.1.5 <code>bessel_k()</code>	59
5.1.1.6 <code>bessel_k_ex()</code>	59
5.1.1.7 <code>bessel_y()</code>	59
5.1.1.8 <code>bessel_y_ex()</code>	60
5.1.1.9 <code>beta()</code>	60
5.1.1.10 <code>choose()</code>	60
5.1.1.11 <code>dbeta()</code>	60
5.1.1.12 <code>dbinom()</code>	61
5.1.1.13 <code>dcauchy()</code>	61
5.1.1.14 <code>dchisq()</code>	61
5.1.1.15 <code>dexp()</code>	62
5.1.1.16 <code>df()</code>	62
5.1.1.17 <code>dgamma()</code>	62
5.1.1.18 <code>dgeom()</code>	62
5.1.1.19 <code>dhyper()</code>	63
5.1.1.20 <code>digamma()</code>	63
5.1.1.21 <code>dlnorm()</code>	63
5.1.1.22 <code>dlogis()</code>	63
5.1.1.23 <code>dnbeta()</code>	64
5.1.1.24 <code>dnbinom()</code>	64
5.1.1.25 <code>dnbinom_mu()</code>	64
5.1.1.26 <code>dnychisq()</code>	64

---

5.1.1.27 dnf()	65
5.1.1.28 dnorm()	65
5.1.1.29 dnt()	66
5.1.1.30 dpois()	66
5.1.1.31 dpsifn()	66
5.1.1.32 dsigrank()	67
5.1.1.33 dt()	67
5.1.1.34 dunif()	67
5.1.1.35 dweibull()	67
5.1.1.36 dwilcox()	68
5.1.1.37 exp_rand()	68
5.1.1.38 expm1()	68
5.1.1.39 fmax2()	68
5.1.1.40 fmin2()	69
5.1.1.41 fprec()	69
5.1.1.42 fround()	69
5.1.1.43 fsign()	69
5.1.1.44 ftrunc()	69
5.1.1.45 gammafn()	70
5.1.1.46 hypot()	70
5.1.1.47 imax2()	70
5.1.1.48 imin2()	70
5.1.1.49 lbeta()	70
5.1.1.50 lchoose()	71
5.1.1.51 lgamma1p()	71
5.1.1.52 lgammafn()	71
5.1.1.53 lgammafn_sign()	71
5.1.1.54 log1p()	71
5.1.1.55 log1pexp()	72
5.1.1.56 log1pmx()	72
5.1.1.57 logspace_add()	72
5.1.1.58 logspace_sub()	72
5.1.1.59 norm_rand()	72
5.1.1.60 pbeta()	73
5.1.1.61 pbinom()	73
5.1.1.62 pcauchy()	73
5.1.1.63 pchisq()	74
5.1.1.64 pentagamma()	74
5.1.1.65 pexp()	74



---

5.1.1.66 pf()	75
5.1.1.67 pgamma()	75
5.1.1.68 pgeom()	75
5.1.1.69 phyper()	75
5.1.1.70 plnorm()	76
5.1.1.71 plogis()	76
5.1.1.72 pnbeta()	76
5.1.1.73 pnbinom()	76
5.1.1.74 pnbinom_mu()	77
5.1.1.75 pnchisq()	77
5.1.1.76 pnf()	77
5.1.1.77 pnorm()	78
5.1.1.78 pnorm_both()	78
5.1.1.79 pnt()	78
5.1.1.80 ppois()	79
5.1.1.81 psigamma()	79
5.1.1.82 psignrank()	79
5.1.1.83 pt()	79
5.1.1.84 ptukey()	80
5.1.1.85 punif()	80
5.1.1.86 pweibull()	80
5.1.1.87 pwilcox()	81
5.1.1.88 qbeta()	81
5.1.1.89 qbinom()	81
5.1.1.90 qcauchy()	81
5.1.1.91 qchisq()	82
5.1.1.92 qexp()	82
5.1.1.93 qf()	82
5.1.1.94 qgamma()	83
5.1.1.95 qgeom()	83
5.1.1.96 qhyper()	83
5.1.1.97 qlnorm()	83
5.1.1.98 qlogis()	84
5.1.1.99 qnbeta()	84
5.1.1.100 qnbinom()	84
5.1.1.101 qnbinom_mu()	84
5.1.1.102 qnchisq()	85
5.1.1.103 qnf()	85
5.1.1.104 qnorm()	85

---

5.1.1.105 qnt()	86
5.1.1.106 qpois()	86
5.1.1.107 qsignrank()	86
5.1.1.108 qt()	87
5.1.1.109 qtukey()	87
5.1.1.110 qunif()	87
5.1.1.111 qweibull()	88
5.1.1.112 qwilcox()	88
5.1.1.113 rbeta()	88
5.1.1.114 rbinom()	88
5.1.1.115 rcauchy()	89
5.1.1.116 rchisq()	89
5.1.1.117 rexp()	89
5.1.1.118 rf()	89
5.1.1.119 rgamma()	90
5.1.1.120 rgeom()	90
5.1.1.121 rhyper()	90
5.1.1.122 rlnorm()	90
5.1.1.123 rlogis()	90
5.1.1.124 rmultinom()	91
5.1.1.125 rnbeta()	91
5.1.1.126 rnbinom()	91
5.1.1.127 rnchisq()	91
5.1.1.128 rnorm()	92
5.1.1.129 rpois()	92
5.1.1.130 rsignrank()	92
5.1.1.131 rt()	92
5.1.1.132 runif()	92
5.1.1.133 rweibull()	93
5.1.1.134 rwilcox()	93
5.1.1.135 sign()	93
5.1.1.136 tetragamma()	93
5.1.1.137 trigamma()	93
5.1.1.138 unif_rand()	94
5.2 Rcpp Namespace Reference	94
5.2.1 Detailed Description	111
5.2.2 Typedef Documentation	112
5.2.2.1 CharacterMatrix	112
5.2.2.2 CharacterVector	112

---

5.2.2.3 ComplexMatrix	112
5.2.2.4 ComplexVector	112
5.2.2.5 const_Proxy	112
5.2.2.6 DataFrame	113
5.2.2.7 DatetimeVector	113
5.2.2.8 DateVector	113
5.2.2.9 DottedPair	113
5.2.2.10 DoubleVector	113
5.2.2.11 Environment	113
5.2.2.12 ExpressionMatrix	114
5.2.2.13 ExpressionVector	114
5.2.2.14 Formula	114
5.2.2.15 Function	114
5.2.2.16 GenericMatrix	114
5.2.2.17 GenericVector	115
5.2.2.18 IntegerMatrix	115
5.2.2.19 IntegerVector	115
5.2.2.20 InternalFunction	115
5.2.2.21 Language	115
5.2.2.22 List	115
5.2.2.23 ListMatrix	116
5.2.2.24 LogicalMatrix	116
5.2.2.25 LogicalVector	116
5.2.2.26 nanotime_t	116
5.2.2.27 NumericMatrix	116
5.2.2.28 NumericVector	116
5.2.2.29 Pairlist	117
5.2.2.30 Promise	117
5.2.2.31 RawMatrix	117
5.2.2.32 RawVector	117
5.2.2.33 Reference	117
5.2.2.34 RObject	117
5.2.2.35 S4	118
5.2.2.36 StretchyList	118
5.2.2.37 StringMatrix	118
5.2.2.38 StringVector	118
5.2.2.39 Symbol	118
5.2.2.40 ValidConstructor	118
5.2.2.41 ValidMethod	119

---

5.2.2.42 WeakReference	119
5.2.3 Function Documentation	119
5.2.3.1 __any() [1/2]	119
5.2.3.2 __any() [2/2]	119
5.2.3.3 __any_if() [1/2]	120
5.2.3.4 __any_if() [2/2]	120
5.2.3.5 all()	120
5.2.3.6 any() [1/2]	120
5.2.3.7 any() [2/2]	121
5.2.3.8 any_if()	121
5.2.3.9 as() [1/2]	122
5.2.3.10 as() [2/2]	122
5.2.3.11 as< char >()	122
5.2.3.12 as_vector()	123
5.2.3.13 assign() [1/3]	123
5.2.3.14 assign() [2/3]	123
5.2.3.15 assign() [3/3]	124
5.2.3.16 bare_as()	125
5.2.3.17 base_env()	125
5.2.3.18 base_namespace()	125
5.2.3.19 bindingsActive()	125
5.2.3.20 bindingsLocked()	126
5.2.3.21 checkUserInterrupt()	127
5.2.3.22 clamp()	127
5.2.3.23 clone()	128
5.2.3.24 col()	128
5.2.3.25 collapse()	128
5.2.3.26 colMeans()	129
5.2.3.27 colnames()	129
5.2.3.28 colSums()	129
5.2.3.29 converter()	129
5.2.3.30 cummax() [1/2]	130
5.2.3.31 cummax() [2/2]	130
5.2.3.32 cummin() [1/2]	130
5.2.3.33 cummin() [2/2]	130
5.2.3.34 cumprod() [1/3]	130
5.2.3.35 cumprod() [2/3]	131
5.2.3.36 cumprod() [3/3]	131
5.2.3.37 cumsum() [1/2]	131

---

5.2.3.38 cumsum() [2/2]	131
5.2.3.39 detzcode()	131
5.2.3.40 detzcode64()	132
5.2.3.41 dexp()	132
5.2.3.42 diag()	132
5.2.3.43 diff() [1/2]	133
5.2.3.44 diff() [2/2]	133
5.2.3.45 differ_by_repeat()	133
5.2.3.46 duplicated()	133
5.2.3.47 empty_env()	134
5.2.3.48 Environment_Impl() [1/3]	134
5.2.3.49 Environment_Impl() [2/3]	134
5.2.3.50 Environment_Impl() [3/3]	134
5.2.3.51 eval() [1/2]	135
5.2.3.52 eval() [2/2]	135
5.2.3.53 exists()	135
5.2.3.54 fast_eval() [1/2]	136
5.2.3.55 fast_eval() [2/2]	137
5.2.3.56 finalizer_wrapper()	137
5.2.3.57 find() [1/2]	137
5.2.3.58 find() [2/2]	138
5.2.3.59 Formula_Impl() [1/2]	139
5.2.3.60 Formula_Impl() [2/2]	139
5.2.3.61 get() [1/2]	139
5.2.3.62 get() [2/2]	140
5.2.3.63 get_return_type()	141
5.2.3.64 get_return_type< Rcpp::CharacterVector >()	141
5.2.3.65 get_return_type< Rcpp::ExpressionVector >()	141
5.2.3.66 get_return_type< Rcpp::IntegerVector >()	142
5.2.3.67 get_return_type< Rcpp::List >()	142
5.2.3.68 get_return_type< Rcpp::NumericVector >()	142
5.2.3.69 get_return_type< Rcpp::RawVector >()	142
5.2.3.70 get_return_type< SEXP >()	142
5.2.3.71 get_return_type< void_type >()	143
5.2.3.72 get_return_type_dispatch() [1/2]	143
5.2.3.73 get_return_type_dispatch() [2/2]	143
5.2.3.74 getnum()	144
5.2.3.75 getoffset()	144
5.2.3.76 getqzname()	145

---

5.2.3.77	<a href="#">getrule()</a>	145
5.2.3.78	<a href="#">getsecs()</a>	145
5.2.3.79	<a href="#">getzname()</a>	146
5.2.3.80	<a href="#">global_env()</a>	146
5.2.3.81	<a href="#">gmtime_()</a>	146
5.2.3.82	<a href="#">gmtload()</a>	147
5.2.3.83	<a href="#">gmtsub()</a>	147
5.2.3.84	<a href="#">grow()</a> [1/3]	148
5.2.3.85	<a href="#">grow()</a> [2/3]	148
5.2.3.86	<a href="#">grow()</a> [3/3]	149
5.2.3.87	<a href="#">head()</a>	150
5.2.3.88	<a href="#">ifelse()</a> [1/7]	150
5.2.3.89	<a href="#">ifelse()</a> [2/7]	150
5.2.3.90	<a href="#">ifelse()</a> [3/7]	150
5.2.3.91	<a href="#">ifelse()</a> [4/7]	151
5.2.3.92	<a href="#">ifelse()</a> [5/7]	151
5.2.3.93	<a href="#">ifelse()</a> [6/7]	151
5.2.3.94	<a href="#">ifelse()</a> [7/7]	151
5.2.3.95	<a href="#">in()</a>	152
5.2.3.96	<a href="#">increment_overflow()</a>	152
5.2.3.97	<a href="#">increment_overflow_time()</a>	152
5.2.3.98	<a href="#">intersect()</a>	153
5.2.3.99	<a href="#">is()</a>	153
5.2.3.100	<a href="#">is_false()</a>	153
5.2.3.101	<a href="#">is_finite()</a>	154
5.2.3.102	<a href="#">is_infinite()</a>	154
5.2.3.103	<a href="#">is_na()</a> [1/6]	155
5.2.3.104	<a href="#">is_na()</a> [2/6]	155
5.2.3.105	<a href="#">is_na()</a> [3/6]	155
5.2.3.106	<a href="#">is_na()</a> [4/6]	156
5.2.3.107	<a href="#">is_na()</a> [5/6]	156
5.2.3.108	<a href="#">is_na()</a> [6/6]	156
5.2.3.109	<a href="#">is_nan()</a>	156
5.2.3.110	<a href="#">is_true()</a>	157
5.2.3.111	<a href="#">is_user_database()</a>	157
5.2.3.112	<a href="#">isLocked()</a>	157
5.2.3.113	<a href="#">Language_Impl()</a> [1/5]	157
5.2.3.114	<a href="#">Language_Impl()</a> [2/5]	157
5.2.3.115	<a href="#">Language_Impl()</a> [3/5]	158

---

5.2.3.116 Language_Impl() [4/5]	158
5.2.3.117 Language_Impl() [5/5]	159
5.2.3.118 lapply() [1/2]	159
5.2.3.119 lapply() [2/2]	159
5.2.3.120 leaps_thru_end_of()	160
5.2.3.121 lock()	160
5.2.3.122 lockBinding()	160
5.2.3.123 lower_tri()	161
5.2.3.124 ls()	161
5.2.3.125 make_string_transformer()	162
5.2.3.126 mapply() [1/4]	162
5.2.3.127 mapply() [2/4]	163
5.2.3.128 mapply() [3/4]	163
5.2.3.129 mapply() [4/4]	164
5.2.3.130 match()	164
5.2.3.131 max()	165
5.2.3.132 mean() [1/4]	165
5.2.3.133 mean() [2/4]	165
5.2.3.134 mean() [3/4]	166
5.2.3.135 mean() [4/4]	166
5.2.3.136 median()	166
5.2.3.137 message()	166
5.2.3.138 min()	167
5.2.3.139 mktime00()	167
5.2.3.140 na_omit()	168
5.2.3.141 Named() [1/2]	168
5.2.3.142 Named() [2/2]	168
5.2.3.143 namespace_env()	168
5.2.3.144 new_child()	169
5.2.3.145 new_env() [1/2]	170
5.2.3.146 new_env() [2/2]	170
5.2.3.147 no_init() [1/2]	170
5.2.3.148 no_init() [2/2]	171
5.2.3.149 noNA() [1/3]	171
5.2.3.150 noNA() [2/3]	171
5.2.3.151 noNA() [3/3]	171
5.2.3.152 operator SEXP()	172
5.2.3.153 operator"!="() [1/4]	172
5.2.3.154 operator"!="() [2/4]	172

---

5.2.3.155 operator"!=( ) [ 3/4 ] . . . . .	172
5.2.3.156 operator"!=( ) [ 4/4 ] . . . . .	172
5.2.3.157 operator*( ) [ 1/5 ] . . . . .	173
5.2.3.158 operator*( ) [ 2/5 ] . . . . .	173
5.2.3.159 operator*( ) [ 3/5 ] . . . . .	173
5.2.3.160 operator*( ) [ 4/5 ] . . . . .	173
5.2.3.161 operator*( ) [ 5/5 ] . . . . .	174
5.2.3.162 operator+( ) [ 1/8 ] . . . . .	174
5.2.3.163 operator+( ) [ 2/8 ] . . . . .	174
5.2.3.164 operator+( ) [ 3/8 ] . . . . .	174
5.2.3.165 operator+( ) [ 4/8 ] . . . . .	175
5.2.3.166 operator+( ) [ 5/8 ] . . . . .	175
5.2.3.167 operator+( ) [ 6/8 ] . . . . .	175
5.2.3.168 operator+( ) [ 7/8 ] . . . . .	175
5.2.3.169 operator+( ) [ 8/8 ] . . . . .	176
5.2.3.170 operator-( ) [ 1/5 ] . . . . .	176
5.2.3.171 operator-( ) [ 2/5 ] . . . . .	176
5.2.3.172 operator-( ) [ 3/5 ] . . . . .	176
5.2.3.173 operator-( ) [ 4/5 ] . . . . .	177
5.2.3.174 operator-( ) [ 5/5 ] . . . . .	177
5.2.3.175 operator/( ) [ 1/3 ] . . . . .	177
5.2.3.176 operator/( ) [ 2/3 ] . . . . .	177
5.2.3.177 operator/( ) [ 3/3 ] . . . . .	178
5.2.3.178 operator<( ) [ 1/2 ] . . . . .	178
5.2.3.179 operator<( ) [ 2/2 ] . . . . .	178
5.2.3.180 operator<<( ) [ 1/10 ] . . . . .	178
5.2.3.181 operator<<( ) [ 2/10 ] . . . . .	179
5.2.3.182 operator<<( ) [ 3/10 ] . . . . .	179
5.2.3.183 operator<<( ) [ 4/10 ] . . . . .	179
5.2.3.184 operator<<( ) [ 5/10 ] . . . . .	179
5.2.3.185 operator<<( ) [ 6/10 ] . . . . .	180
5.2.3.186 operator<<( ) [ 7/10 ] . . . . .	181
5.2.3.187 operator<<( ) [ 8/10 ] . . . . .	181
5.2.3.188 operator<<( ) [ 9/10 ] . . . . .	182
5.2.3.189 operator<<( ) [ 10/10 ] . . . . .	182
5.2.3.190 operator<=( ) [ 1/2 ] . . . . .	182
5.2.3.191 operator<=( ) [ 2/2 ] . . . . .	182
5.2.3.192 operator==( ) [ 1/4 ] . . . . .	183
5.2.3.193 operator==( ) [ 2/4 ] . . . . .	183



---

5.2.3.194 operator==( ) [3/4]	183
5.2.3.195 operator==( ) [4/4]	183
5.2.3.196 operator>( ) [1/2]	183
5.2.3.197 operator>( ) [2/2]	184
5.2.3.198 operator>=( ) [1/2]	184
5.2.3.199 operator>=( ) [2/2]	184
5.2.3.200 outer( )	184
5.2.3.201 pairlist( )	185
5.2.3.202 Pairlist_Impl( ) [1/2]	185
5.2.3.203 Pairlist_Impl( ) [2/2]	185
5.2.3.204 parent( )	185
5.2.3.205 pexp( )	186
5.2.3.206 pmax( ) [1/3]	186
5.2.3.207 pmax( ) [2/3]	187
5.2.3.208 pmax( ) [3/3]	187
5.2.3.209 pmin( ) [1/3]	188
5.2.3.210 pmin( ) [2/3]	188
5.2.3.211 pmin( ) [3/3]	189
5.2.3.212 pow( )	189
5.2.3.213 print( )	190
5.2.3.214 push_back( )	190
5.2.3.215 push_back__impl( ) [1/2]	190
5.2.3.216 push_back__impl( ) [2/2]	191
5.2.3.217 push_front( )	191
5.2.3.218 push_front__impl( ) [1/2]	191
5.2.3.219 push_front__impl( ) [2/2]	192
5.2.3.220 qexp( )	192
5.2.3.221 r_cast( )	192
5.2.3.222 range( )	193
5.2.3.223 rbeta( )	193
5.2.3.224 rbinom( )	194
5.2.3.225 rcauchy( ) [1/3]	194
5.2.3.226 rcauchy( ) [2/3]	194
5.2.3.227 rcauchy( ) [3/3]	194
5.2.3.228 rchisq( )	195
5.2.3.229 RCPP_API_CLASS( ) [1/14]	195
5.2.3.230 RCPP_API_CLASS( ) [2/14]	195
5.2.3.231 RCPP_API_CLASS( ) [3/14]	195
5.2.3.232 RCPP_API_CLASS( ) [4/14]	195

---

5.2.3.233 RCPP_API_CLASS() [5/14]	196
5.2.3.234 RCPP_API_CLASS() [6/14]	197
5.2.3.235 RCPP_API_CLASS() [7/14]	197
5.2.3.236 RCPP_API_CLASS() [8/14]	197
5.2.3.237 RCPP_API_CLASS() [9/14]	198
5.2.3.238 RCPP_API_CLASS() [10/14]	198
5.2.3.239 RCPP_API_CLASS() [11/14]	198
5.2.3.240 RCPP_API_CLASS() [12/14]	199
5.2.3.241 RCPP_API_CLASS() [13/14]	200
5.2.3.242 RCPP_API_CLASS() [14/14]	200
5.2.3.243 Rcpp_cerr_get()	201
5.2.3.244 Rcpp_cout_get()	201
5.2.3.245 Rcpp_eval()	202
5.2.3.246 RCPP_EXCEPTION_CLASS()	202
5.2.3.247 Rcpp_fast_eval()	202
5.2.3.248 Rcpp_lang10()	203
5.2.3.249 Rcpp_lang11()	203
5.2.3.250 Rcpp_lang12()	204
5.2.3.251 Rcpp_lang13()	205
5.2.3.252 Rcpp_lang14()	205
5.2.3.253 Rcpp_lang15()	206
5.2.3.254 Rcpp_lang16()	206
5.2.3.255 Rcpp_lang17()	207
5.2.3.256 Rcpp_lang18()	208
5.2.3.257 Rcpp_lang19()	208
5.2.3.258 Rcpp_lang20()	209
5.2.3.259 Rcpp_lang7()	210
5.2.3.260 Rcpp_lang8()	210
5.2.3.261 Rcpp_lang9()	211
5.2.3.262 Rcpp_list10()	211
5.2.3.263 Rcpp_list11()	212
5.2.3.264 Rcpp_list12()	212
5.2.3.265 Rcpp_list13()	213
5.2.3.266 Rcpp_list14()	213
5.2.3.267 Rcpp_list15()	214
5.2.3.268 Rcpp_list16()	215
5.2.3.269 Rcpp_list17()	215
5.2.3.270 Rcpp_list18()	216
5.2.3.271 Rcpp_list19()	217

---

5.2.3.272 Rcpp_list2()	217
5.2.3.273 Rcpp_list20()	218
5.2.3.274 Rcpp_list3()	218
5.2.3.275 Rcpp_list4()	219
5.2.3.276 Rcpp_list5()	220
5.2.3.277 Rcpp_list6()	220
5.2.3.278 Rcpp_list7()	221
5.2.3.279 Rcpp_list8()	221
5.2.3.280 Rcpp_list9()	222
5.2.3.281 Rcpp_namespace()	222
5.2.3.282 Rcpp_precious_init()	223
5.2.3.283 Rcpp_precious_preserve()	223
5.2.3.284 Rcpp_precious_remove()	223
5.2.3.285 Rcpp_precious_teardown()	223
5.2.3.286 Rcpp_PreciousPreserve()	224
5.2.3.287 Rcpp_PreciousRelease()	224
5.2.3.288 Rcpp_PreserveObject()	225
5.2.3.289 Rcpp_protect()	225
5.2.3.290 Rcpp_ReleaseObject()	225
5.2.3.291 Rcpp_ReplaceObject()	225
5.2.3.292 RCPP_SIMPLE_EXCEPTION_CLASS()	226
5.2.3.293 Rcpp_unprotect()	226
5.2.3.294 remove()	226
5.2.3.295 rep() [1/6]	226
5.2.3.296 rep() [2/6]	227
5.2.3.297 rep() [3/6]	227
5.2.3.298 rep() [4/6]	227
5.2.3.299 rep() [5/6]	227
5.2.3.300 rep() [6/6]	227
5.2.3.301 rep_each()	228
5.2.3.302 rep_len()	228
5.2.3.303 rev()	228
5.2.3.304 rexp() [1/2]	228
5.2.3.305 rexp() [2/2]	228
5.2.3.306 rf()	229
5.2.3.307 rgamma() [1/2]	229
5.2.3.308 rgamma() [2/2]	229
5.2.3.309 rgeom()	229
5.2.3.310 rhyper()	230

---

5.2.3.311 rlnorm() [1/3]	230
5.2.3.312 rlnorm() [2/3]	230
5.2.3.313 rlnorm() [3/3]	231
5.2.3.314 rlogis() [1/3]	231
5.2.3.315 rlogis() [2/3]	231
5.2.3.316 rlogis() [3/3]	232
5.2.3.317 rbinom()	232
5.2.3.318 rbinom_mu()	232
5.2.3.319 rchisq() [1/2]	232
5.2.3.320 rchisq() [2/2]	233
5.2.3.321 rnorm() [1/3]	233
5.2.3.322 rnorm() [2/3]	233
5.2.3.323 rnorm() [3/3]	234
5.2.3.324 row()	234
5.2.3.325 rowMeans()	234
5.2.3.326 rownames()	235
5.2.3.327 rowSums()	235
5.2.3.328 rpois()	235
5.2.3.329 rsignrank()	235
5.2.3.330 rt()	236
5.2.3.331 runif() [1/3]	236
5.2.3.332 runif() [2/3]	236
5.2.3.333 runif() [3/3]	237
5.2.3.334 rweibull() [1/2]	237
5.2.3.335 rweibull() [2/2]	237
5.2.3.336 rwilcox()	238
5.2.3.337 sample() [1/2]	238
5.2.3.338 sample() [2/2]	239
5.2.3.339 sapply() [1/2]	239
5.2.3.340 sapply() [2/2]	240
5.2.3.341 sd()	240
5.2.3.342 self_match()	241
5.2.3.343 seq()	241
5.2.3.344 seq_along()	241
5.2.3.345 seq_len()	242
5.2.3.346 setdiff()	242
5.2.3.347 setequal()	242
5.2.3.348 setFunction()	243
5.2.3.349 setSymbol() [1/2]	243

---

5.2.3.350 setSymbol() [2/2]	244
5.2.3.351 shush_about_NA()	244
5.2.3.352 sign() [1/2]	244
5.2.3.353 sign() [2/2]	245
5.2.3.354 sort_unique()	245
5.2.3.355 sprintf()	245
5.2.3.356 standard_delete_finalizer()	246
5.2.3.357 stop() [1/12]	246
5.2.3.358 stop() [2/12]	247
5.2.3.359 stop() [3/12]	247
5.2.3.360 stop() [4/12]	248
5.2.3.361 stop() [5/12]	248
5.2.3.362 stop() [6/12]	249
5.2.3.363 stop() [7/12]	249
5.2.3.364 stop() [8/12]	250
5.2.3.365 stop() [9/12]	250
5.2.3.366 stop() [10/12]	251
5.2.3.367 stop() [11/12]	252
5.2.3.368 stop() [12/12]	252
5.2.3.369 StretchyList_Impl() [1/2]	253
5.2.3.370 StretchyList_Impl() [2/2]	253
5.2.3.371 sum() [1/3]	253
5.2.3.372 sum() [2/3]	253
5.2.3.373 sum() [3/3]	253
5.2.3.374 table()	254
5.2.3.375 tail()	254
5.2.3.376 timesub()	255
5.2.3.377 toString()	255
5.2.3.378 tranpose_impl()	255
5.2.3.379 transpose() [1/3]	256
5.2.3.380 transpose() [2/3]	256
5.2.3.381 transpose() [3/3]	256
5.2.3.382 transtime()	256
5.2.3.383 trimws() [1/3]	257
5.2.3.384 trimws() [2/3]	257
5.2.3.385 trimws() [3/3]	258
5.2.3.386 type2name()	258
5.2.3.387 typesequiv()	258
5.2.3.388 tzload()	259

---

5.2.3.389	<a href="#">tzparse()</a>	259
5.2.3.390	<a href="#">union_()</a>	260
5.2.3.391	<a href="#">unique()</a>	261
5.2.3.392	<a href="#">unlockBinding()</a>	261
5.2.3.393	<a href="#">unwindProtect()</a>	262
5.2.3.394	<a href="#">update()</a>	262
5.2.3.395	<a href="#">upper_tri()</a>	263
5.2.3.396	<a href="#">var()</a> [1/4]	263
5.2.3.397	<a href="#">var()</a> [2/4]	263
5.2.3.398	<a href="#">var()</a> [3/4]	264
5.2.3.399	<a href="#">var()</a> [4/4]	264
5.2.3.400	<a href="#">warning()</a> [1/12]	264
5.2.3.401	<a href="#">warning()</a> [2/12]	265
5.2.3.402	<a href="#">warning()</a> [3/12]	265
5.2.3.403	<a href="#">warning()</a> [4/12]	266
5.2.3.404	<a href="#">warning()</a> [5/12]	266
5.2.3.405	<a href="#">warning()</a> [6/12]	267
5.2.3.406	<a href="#">warning()</a> [7/12]	267
5.2.3.407	<a href="#">warning()</a> [8/12]	268
5.2.3.408	<a href="#">warning()</a> [9/12]	268
5.2.3.409	<a href="#">warning()</a> [10/12]	269
5.2.3.410	<a href="#">warning()</a> [11/12]	270
5.2.3.411	<a href="#">warning()</a> [12/12]	270
5.2.3.412	<a href="#">warningcall()</a>	271
5.2.3.413	<a href="#">which_max()</a>	271
5.2.3.414	<a href="#">which_min()</a>	271
5.2.3.415	<a href="#">wrap()</a>	272
5.2.3.416	<a href="#">wrap&lt; Datetime &gt;()</a>	273
5.2.3.417	<a href="#">wrap&lt; Rcpp::Date &gt;()</a>	273
5.2.3.418	<a href="#">wrap&lt; Rcpp::Datetime &gt;()</a>	273
5.2.3.419	<a href="#">wrap&lt; Rcpp::String &gt;()</a>	273
5.2.3.420	<a href="#">wrap_extra_steps()</a>	274
5.2.3.421	<a href="#">wrap_extra_steps&lt; Rcpp::Date &gt;()</a>	274
5.2.3.422	<a href="#">wrap_extra_steps&lt; Rcpp::Datetime &gt;()</a>	274
5.2.4	Variable Documentation	274
5.2.4.1	<a href="#">_</a>	275
5.2.4.2	<a href="#">BindingPolicy&lt; Environment_Impl&lt; StoragePolicy &gt; &gt;</a>	275
5.2.4.3	<a href="#">current_scope</a>	275
5.2.4.4	<a href="#">DottedPairImpl&lt; DottedPair_Impl&lt; StoragePolicy &gt; &gt;</a>	276

5.2.4.5 DottedPairImpl< Formula_Impl< StoragePolicy > > . . . . .	276
5.2.4.6 DottedPairImpl< Language_Impl< StoragePolicy > > . . . . .	276
5.2.4.7 DottedPairImpl< Pairlist_Impl< StoragePolicy > > . . . . .	276
5.2.4.8 DottedPairProxyPolicy< DottedPair_Impl< StoragePolicy > > . . . . .	277
5.2.4.9 DottedPairProxyPolicy< Formula_Impl< StoragePolicy > > . . . . .	277
5.2.4.10 DottedPairProxyPolicy< Language_Impl< StoragePolicy > > . . . . .	277
5.2.4.11 DottedPairProxyPolicy< Pairlist_Impl< StoragePolicy > > . . . . .	277
5.2.4.12 DottedPairProxyPolicy< StretchyList_Impl< StoragePolicy > > . . . . .	277
5.2.4.13 FieldProxyPolicy< Reference_Impl< StoragePolicy > > . . . . .	278
5.2.4.14 gmt . . . . .	278
5.2.4.15 gmt_is_set . . . . .	278
5.2.4.16 gmtmem . . . . .	278
5.2.4.17 mon_lengths . . . . .	279
5.2.4.18 NA . . . . .	279
5.2.4.19 Rcerr . . . . .	279
5.2.4.20 Rcout . . . . .	279
5.2.4.21 Rcpp_precious . . . . .	280
5.2.4.22 time_t_max . . . . .	280
5.2.4.23 time_t_min . . . . .	280
5.2.4.24 tm . . . . .	280
5.2.4.25 year_lengths . . . . .	280
5.3 Rcpp::algorithm Namespace Reference . . . . .	281
5.3.1 Function Documentation . . . . .	282
5.3.1.1 exp() . . . . .	282
5.3.1.2 log() . . . . .	282
5.3.1.3 max() . . . . .	283
5.3.1.4 max_nona() . . . . .	283
5.3.1.5 mean() [1/2] . . . . .	284
5.3.1.6 mean() [2/2] . . . . .	284
5.3.1.7 min() . . . . .	285
5.3.1.8 min_nona() . . . . .	285
5.3.1.9 prod() . . . . .	286
5.3.1.10 prod_nona() . . . . .	286
5.3.1.11 sqrt() . . . . .	286
5.3.1.12 sum() . . . . .	287
5.3.1.13 sum_nona() . . . . .	287
5.4 Rcpp::algorithm::helpers Namespace Reference . . . . .	287
5.5 Rcpp::attributes Namespace Reference . . . . .	288
5.5.1 Function Documentation . . . . .	289

---

5.5.1.1	checkRSignature()	290
5.5.1.2	createDirectory()	290
5.5.1.3	endsWith()	291
5.5.1.4	generateCpp()	291
5.5.1.5	generateRArgList()	292
5.5.1.6	initializeGlobals()	292
5.5.1.7	isQuoted()	292
5.5.1.8	isRoxygenCpp()	293
5.5.1.9	isWhitespace()	293
5.5.1.10	operator<<() [1/5]	293
5.5.1.11	operator<<() [2/5]	294
5.5.1.12	operator<<() [3/5]	294
5.5.1.13	operator<<() [4/5]	295
5.5.1.14	operator<<() [5/5]	295
5.5.1.15	printArgument()	296
5.5.1.16	printFunction()	296
5.5.1.17	removeFile()	297
5.5.1.18	showWarning()	297
5.5.1.19	stripQuotes()	298
5.5.1.20	stripTrailingLineComments()	298
5.5.1.21	trimWhitespace()	298
5.5.2	Variable Documentation	298
5.5.2.1	kDependsAttribute	299
5.5.2.2	kExportAttribute	299
5.5.2.3	kExportInvisible	299
5.5.2.4	kExportName	299
5.5.2.5	kExportRng	299
5.5.2.6	kExportSignature	300
5.5.2.7	kInitAttribute	300
5.5.2.8	kInterfaceCpp	300
5.5.2.9	kInterfaceR	300
5.5.2.10	kInterfacesAttribute	301
5.5.2.11	kParamBlockEnd	301
5.5.2.12	kParamBlockStart	301
5.5.2.13	kParamValueFalse	301
5.5.2.14	kParamValueFALSE	301
5.5.2.15	kParamValueTrue	302
5.5.2.16	kParamValueTRUE	302
5.5.2.17	kPluginsAttribute	302



---

5.5.2.18 kWhitespaceChars	302
5.6 Rcpp::internal Namespace Reference	302
5.6.1 Detailed Description	307
5.6.2 Function Documentation	307
5.6.2.1 as() [1/10]	307
5.6.2.2 as() [2/10]	307
5.6.2.3 as() [3/10]	308
5.6.2.4 as() [4/10]	308
5.6.2.5 as() [5/10]	308
5.6.2.6 as() [6/10]	309
5.6.2.7 as() [7/10]	309
5.6.2.8 as() [8/10]	309
5.6.2.9 as() [9/10]	309
5.6.2.10 as() [10/10]	310
5.6.2.11 as_module_object()	310
5.6.2.12 as_module_object_internal()	310
5.6.2.13 as_string() [1/2]	311
5.6.2.14 as_string() [2/2]	311
5.6.2.15 as_vector__impl() [1/2]	312
5.6.2.16 as_vector__impl() [2/2]	312
5.6.2.17 basic_cast()	313
5.6.2.18 beginSuspendRNGSynchronization()	313
5.6.2.19 caster< double, Rcpp::Date >()	313
5.6.2.20 caster< double, Rcpp::Datetime >()	313
5.6.2.21 caster< Rcpp::Date, double >()	314
5.6.2.22 caster< Rcpp::Datetime, double >()	314
5.6.2.23 check_single_string()	315
5.6.2.24 complex__acos()	315
5.6.2.25 complex__acosh()	316
5.6.2.26 complex__Arg()	316
5.6.2.27 complex__asin()	316
5.6.2.28 complex__asinh()	317
5.6.2.29 complex__atan()	318
5.6.2.30 complex__atanh()	318
5.6.2.31 complex__Conj()	319
5.6.2.32 complex__cos()	319
5.6.2.33 complex__cosh()	319
5.6.2.34 complex__exp()	319
5.6.2.35 complex__Im()	320

---

5.6.2.36	<code>complex__log()</code>	320
5.6.2.37	<code>complex__Mod()</code>	320
5.6.2.38	<code>complex__Re()</code>	321
5.6.2.39	<code>complex__sin()</code>	321
5.6.2.40	<code>complex__sinh()</code>	321
5.6.2.41	<code>complex__sqrt()</code>	322
5.6.2.42	<code>complex__tan()</code>	322
5.6.2.43	<code>complex__tanh()</code>	322
5.6.2.44	<code>convert_using_rfunction()</code>	323
5.6.2.45	<code>empty_data_frame()</code>	323
5.6.2.46	<code>endSuspendRNGSynchronization()</code>	324
5.6.2.47	<code>enterRNGScope()</code>	324
5.6.2.48	<code>exitRNGScope()</code>	324
5.6.2.49	<code>export_range__dispatch()</code>	325
5.6.2.50	<code>factorial()</code>	325
5.6.2.51	<code>get_column()</code> [1/2]	325
5.6.2.52	<code>get_column()</code> [2/2]	326
5.6.2.53	<code>get_converter_name()</code>	326
5.6.2.54	<code>get_line()</code>	326
5.6.2.55	<code>get_Rcpp_namespace()</code>	326
5.6.2.56	<code>get_string_buffer()</code>	327
5.6.2.57	<code>getLongjumpToken()</code>	327
5.6.2.58	<code>getPosixClasses()</code>	327
5.6.2.59	<code>grow__dispatch()</code> [1/2]	327
5.6.2.60	<code>grow__dispatch()</code> [2/2]	328
5.6.2.61	<code>interruptedError()</code>	329
5.6.2.62	<code>is__dispatch()</code> [1/2]	329
5.6.2.63	<code>is__dispatch()</code> [2/2]	329
5.6.2.64	<code>is__module__object()</code>	329
5.6.2.65	<code>is__simple()</code>	330
5.6.2.66	<code>is__simple&lt; bool &gt;()</code>	330
5.6.2.67	<code>is__simple&lt; CharacterMatrix &gt;()</code>	330
5.6.2.68	<code>is__simple&lt; CharacterVector &gt;()</code>	331
5.6.2.69	<code>is__simple&lt; ComplexMatrix &gt;()</code>	331
5.6.2.70	<code>is__simple&lt; ComplexVector &gt;()</code>	331
5.6.2.71	<code>is__simple&lt; DataFrame &gt;()</code>	332
5.6.2.72	<code>is__simple&lt; Date &gt;()</code>	332
5.6.2.73	<code>is__simple&lt; Datetime &gt;()</code>	332
5.6.2.74	<code>is__simple&lt; DatetimeVector &gt;()</code>	333

---

5.6.2.75	<code>is__simple&lt; DateVector &gt;()</code>	333
5.6.2.76	<code>is__simple&lt; DottedPair &gt;()</code>	333
5.6.2.77	<code>is__simple&lt; double &gt;()</code>	334
5.6.2.78	<code>is__simple&lt; Environment &gt;()</code>	334
5.6.2.79	<code>is__simple&lt; Formula &gt;()</code>	334
5.6.2.80	<code>is__simple&lt; Function &gt;()</code>	335
5.6.2.81	<code>is__simple&lt; GenericMatrix &gt;()</code>	335
5.6.2.82	<code>is__simple&lt; int &gt;()</code>	335
5.6.2.83	<code>is__simple&lt; IntegerMatrix &gt;()</code>	336
5.6.2.84	<code>is__simple&lt; IntegerVector &gt;()</code>	336
5.6.2.85	<code>is__simple&lt; Language &gt;()</code>	337
5.6.2.86	<code>is__simple&lt; List &gt;()</code>	337
5.6.2.87	<code>is__simple&lt; LogicalMatrix &gt;()</code>	337
5.6.2.88	<code>is__simple&lt; LogicalVector &gt;()</code>	338
5.6.2.89	<code>is__simple&lt; NumericMatrix &gt;()</code>	338
5.6.2.90	<code>is__simple&lt; NumericVector &gt;()</code>	338
5.6.2.91	<code>is__simple&lt; Pairlist &gt;()</code>	339
5.6.2.92	<code>is__simple&lt; Promise &gt;()</code>	339
5.6.2.93	<code>is__simple&lt; RawMatrix &gt;()</code>	339
5.6.2.94	<code>is__simple&lt; RawVector &gt;()</code>	340
5.6.2.95	<code>is__simple&lt; Rcomplex &gt;()</code>	340
5.6.2.96	<code>is__simple&lt; Reference &gt;()</code>	340
5.6.2.97	<code>is__simple&lt; RObject &gt;()</code>	341
5.6.2.98	<code>is__simple&lt; S4 &gt;()</code>	341
5.6.2.99	<code>is__simple&lt; std::string &gt;()</code>	341
5.6.2.100	<code>is__simple&lt; String &gt;()</code>	342
5.6.2.101	<code>is__simple&lt; Symbol &gt;()</code>	342
5.6.2.102	<code>is__simple&lt; WeakReference &gt;()</code>	342
5.6.2.103	<code>is_atomic()</code>	343
5.6.2.104	<code>is_matrix()</code>	343
5.6.2.105	<code>is_module_object_internal()</code>	343
5.6.2.106	<code>is_Rcpp_eval_call()</code>	344
5.6.2.107	<code>isLongjumpSentinel()</code>	344
5.6.2.108	<code>lfactorial()</code>	344
5.6.2.109	<code>longjumpSentinel()</code>	345
5.6.2.110	<code>make_charsexp&lt; Rcpp::String &gt;()</code>	345
5.6.2.111	<code>make_new_object()</code>	345
5.6.2.112	<code>maybeJump()</code>	346
5.6.2.113	<code>new_date_object()</code>	346

---

5.6.2.114 new_posixt_object()	346
5.6.2.115 nth()	347
5.6.2.116 operator+() [1/2]	347
5.6.2.117 operator+() [2/2]	347
5.6.2.118 operator<() [1/2]	347
5.6.2.119 operator<() [2/2]	348
5.6.2.120 operator<<() [1/2]	348
5.6.2.121 operator<<() [2/2]	348
5.6.2.122 operator<=() [1/2]	349
5.6.2.123 operator<=() [2/2]	349
5.6.2.124 operator>() [1/2]	350
5.6.2.125 operator>() [2/2]	350
5.6.2.126 operator>=() [1/2]	351
5.6.2.127 operator>=() [2/2]	351
5.6.2.128 primitive_as()	352
5.6.2.129 r_true_cast()	352
5.6.2.130 r_true_cast< CPLXSXP >()	352
5.6.2.131 r_true_cast< EXPRSXP >()	352
5.6.2.132 r_true_cast< INTSXP >()	353
5.6.2.133 r_true_cast< LANGSXP >()	353
5.6.2.134 r_true_cast< LGLSXP >()	353
5.6.2.135 r_true_cast< LISTSXP >()	354
5.6.2.136 r_true_cast< RAWSXP >()	354
5.6.2.137 r_true_cast< REALSXP >()	354
5.6.2.138 r_true_cast< STRSXP >()	355
5.6.2.139 r_true_cast< VECSXP >()	355
5.6.2.140 range_wrap_dispatch___impl__pair() [1/2]	356
5.6.2.141 range_wrap_dispatch___impl__pair() [2/2]	356
5.6.2.142 Rcpp_eval_impl()	357
5.6.2.143 resumeJump()	357
5.6.2.144 vector_from_string()	357
5.6.2.145 vector_from_string< EXPRSXP >()	358
5.6.2.146 vector_from_string_expr()	358
5.6.2.147 wrap_range_sugar_expression()	358
5.6.3 Variable Documentation	358
5.6.3.1 rngSynchronizationSuspended	358
5.7 Rcpp::internal::debug Namespace Reference	358
5.7.1 Function Documentation	359
5.7.1.1 short_file_name()	359

---

5.8 Rcpp::InternalFunctionWithStdFunction Namespace Reference	359
5.9 Rcpp::stats Namespace Reference	359
5.9.1 Function Documentation	361
5.9.1.1 d_exp_0()	361
5.9.1.2 dcauchy_0()	362
5.9.1.3 dcauchy_1()	362
5.9.1.4 dgamma_1()	362
5.9.1.5 dlnorm_0()	363
5.9.1.6 dlnorm_1()	363
5.9.1.7 dlogis_0()	364
5.9.1.8 dlogis_1()	364
5.9.1.9 dnorm_0()	365
5.9.1.10 dnorm_1()	365
5.9.1.11 dunif_0()	366
5.9.1.12 dunif_1()	366
5.9.1.13 dweibull_1()	366
5.9.1.14 p_exp_0()	367
5.9.1.15 pcauchy_0()	367
5.9.1.16 pcauchy_1()	368
5.9.1.17 pgamma_1()	368
5.9.1.18 plnorm_0()	368
5.9.1.19 plnorm_1()	369
5.9.1.20 plogis_0()	369
5.9.1.21 plogis_1()	370
5.9.1.22 pnorm_0()	370
5.9.1.23 pnorm_1()	371
5.9.1.24 punif_0()	371
5.9.1.25 punif_1()	371
5.9.1.26 pweibull_1()	372
5.9.1.27 q_exp_0()	372
5.9.1.28 qcauchy_0()	373
5.9.1.29 qcauchy_1()	373
5.9.1.30 qgamma_1()	373
5.9.1.31 qlnorm_0()	373
5.9.1.32 qlnorm_1()	374
5.9.1.33 qlogis_0()	374
5.9.1.34 qlogis_1()	375
5.9.1.35 qnorm_0()	376
5.9.1.36 qnorm_1()	376

---

5.9.1.37	qunif_0()	376
5.9.1.38	qunif_1()	376
5.9.1.39	qweibull_1()	377
5.10	Rcpp::sugar Namespace Reference	377
5.10.1	Typedef Documentation	391
5.10.1.1	DDFun	391
5.10.1.2	probs_t	391
5.10.2	Function Documentation	391
5.10.2.1	cbind() [1/49]	391
5.10.2.2	cbind() [2/49]	391
5.10.2.3	cbind() [3/49]	392
5.10.2.4	cbind() [4/49]	392
5.10.2.5	cbind() [5/49]	392
5.10.2.6	cbind() [6/49]	393
5.10.2.7	cbind() [7/49]	393
5.10.2.8	cbind() [8/49]	393
5.10.2.9	cbind() [9/49]	394
5.10.2.10	cbind() [10/49]	394
5.10.2.11	cbind() [11/49]	395
5.10.2.12	cbind() [12/49]	395
5.10.2.13	cbind() [13/49]	396
5.10.2.14	cbind() [14/49]	396
5.10.2.15	cbind() [15/49]	397
5.10.2.16	cbind() [16/49]	397
5.10.2.17	cbind() [17/49]	398
5.10.2.18	cbind() [18/49]	398
5.10.2.19	cbind() [19/49]	399
5.10.2.20	cbind() [20/49]	400
5.10.2.21	cbind() [21/49]	400
5.10.2.22	cbind() [22/49]	401
5.10.2.23	cbind() [23/49]	402
5.10.2.24	cbind() [24/49]	402
5.10.2.25	cbind() [25/49]	403
5.10.2.26	cbind() [26/49]	404
5.10.2.27	cbind() [27/49]	405
5.10.2.28	cbind() [28/49]	405
5.10.2.29	cbind() [29/49]	406
5.10.2.30	cbind() [30/49]	407
5.10.2.31	cbind() [31/49]	408

---

5.10.2.32 <code>cbind()</code> [32/49]	409
5.10.2.33 <code>cbind()</code> [33/49]	410
5.10.2.34 <code>cbind()</code> [34/49]	411
5.10.2.35 <code>cbind()</code> [35/49]	412
5.10.2.36 <code>cbind()</code> [36/49]	413
5.10.2.37 <code>cbind()</code> [37/49]	414
5.10.2.38 <code>cbind()</code> [38/49]	415
5.10.2.39 <code>cbind()</code> [39/49]	416
5.10.2.40 <code>cbind()</code> [40/49]	417
5.10.2.41 <code>cbind()</code> [41/49]	418
5.10.2.42 <code>cbind()</code> [42/49]	419
5.10.2.43 <code>cbind()</code> [43/49]	420
5.10.2.44 <code>cbind()</code> [44/49]	422
5.10.2.45 <code>cbind()</code> [45/49]	423
5.10.2.46 <code>cbind()</code> [46/49]	424
5.10.2.47 <code>cbind()</code> [47/49]	425
5.10.2.48 <code>cbind()</code> [48/49]	426
5.10.2.49 <code>cbind()</code> [49/49]	428
5.10.2.50 <code>collapse__impl()</code>	429
5.10.2.51 <code>EmpiricalSample()</code> [1/2]	429
5.10.2.52 <code>EmpiricalSample()</code> [2/2]	430
5.10.2.53 <code>get_const_begin()</code> [1/2]	431
5.10.2.54 <code>get_const_begin()</code> [2/2]	431
5.10.2.55 <code>get_const_begin__impl()</code> [1/2]	432
5.10.2.56 <code>get_const_begin__impl()</code> [2/2]	432
5.10.2.57 <code>get_const_end()</code>	432
5.10.2.58 <code>na_omit_impl()</code> [1/2]	433
5.10.2.59 <code>na_omit_impl()</code> [2/2]	433
5.10.2.60 <code>Normalize()</code>	434
5.10.2.61 <code>SampleNoReplace()</code> [1/2]	435
5.10.2.62 <code>SampleNoReplace()</code> [2/2]	435
5.10.2.63 <code>SampleReplace()</code> [1/2]	436
5.10.2.64 <code>SampleReplace()</code> [2/2]	437
5.10.2.65 <code>WalkerSample()</code> [1/2]	438
5.10.2.66 <code>WalkerSample()</code> [2/2]	438
5.11 <code>Rcpp::sugar::cbind_impl</code> Namespace Reference	439
5.11.1 Function Documentation	440
5.11.1.1 <code>MakeContainerBindable()</code> [1/4]	440
5.11.1.2 <code>MakeContainerBindable()</code> [2/4]	440

---

5.11.1.3 MakeContainerBindable() [3/4]	441
5.11.1.4 MakeContainerBindable() [4/4]	441
5.11.1.5 MakeScalarBindable()	441
5.11.1.6 operator,()	441
5.12 Rcpp::sugar::cbind_impl::detail Namespace Reference	442
5.12.1 Function Documentation	442
5.12.1.1 MakeBindable() [1/2]	442
5.12.1.2 MakeBindable() [2/2]	442
5.13 Rcpp::sugar::detail Namespace Reference	443
5.13.1 Function Documentation	443
5.13.1.1 check_na() [1/5]	443
5.13.1.2 check_na() [2/5]	444
5.13.1.3 check_na() [3/5]	444
5.13.1.4 check_na() [4/5]	444
5.13.1.5 check_na() [5/5]	444
5.13.1.6 div() [1/2]	444
5.13.1.7 div() [2/2]	445
5.13.1.8 incr() [1/3]	445
5.13.1.9 incr() [2/3]	445
5.13.1.10 incr() [3/3]	445
5.13.1.11 isws()	446
5.13.1.12 set_nan() [1/2]	446
5.13.1.13 set_nan() [2/2]	446
5.13.1.14 trim_both()	446
5.13.1.15 trim_left()	447
5.13.1.16 trim_right()	447
5.14 Rcpp::sugar::median_detail Namespace Reference	448
5.14.1 Function Documentation	448
5.14.1.1 half() [1/3]	448
5.14.1.2 half() [2/3]	448
5.14.1.3 half() [3/3]	449
5.14.1.4 less()	449
5.14.1.5 less< Rcomplex >()	449
5.15 Rcpp::traits Namespace Reference	449
5.15.1 Detailed Description	456
5.15.2 Typedef Documentation	456
5.15.2.1 false_type	456
5.15.2.2 true_type	456
5.15.3 Function Documentation	456



---

5.15.3.1	<a href="#">get_na()</a>	456
5.15.3.2	<a href="#">get_na&lt; CPLXSPX &gt;()</a>	457
5.15.3.3	<a href="#">get_na&lt; INTSPX &gt;()</a>	457
5.15.3.4	<a href="#">get_na&lt; LGLSPX &gt;()</a>	457
5.15.3.5	<a href="#">get_na&lt; REALSPX &gt;()</a>	457
5.15.3.6	<a href="#">get_na&lt; STRSPX &gt;()</a>	457
5.15.3.7	<a href="#">get_na&lt; VECSPX &gt;()</a>	458
5.15.3.8	<a href="#">is_finite()</a>	458
5.15.3.9	<a href="#">is_finite&lt; CPLXSPX &gt;()</a>	458
5.15.3.10	<a href="#">is_finite&lt; INTSPX &gt;()</a>	458
5.15.3.11	<a href="#">is_finite&lt; LGLSPX &gt;()</a>	458
5.15.3.12	<a href="#">is_finite&lt; REALSPX &gt;()</a>	459
5.15.3.13	<a href="#">is_finite&lt; STRSPX &gt;()</a>	459
5.15.3.14	<a href="#">is_infinite()</a>	459
5.15.3.15	<a href="#">is_infinite&lt; CPLXSPX &gt;()</a>	459
5.15.3.16	<a href="#">is_infinite&lt; REALSPX &gt;()</a>	460
5.15.3.17	<a href="#">is_na()</a>	460
5.15.3.18	<a href="#">is_na&lt; CPLXSPX &gt;()</a>	460
5.15.3.19	<a href="#">is_na&lt; INTSPX &gt;()</a>	460
5.15.3.20	<a href="#">is_na&lt; LGLSPX &gt;()</a>	461
5.15.3.21	<a href="#">is_na&lt; REALSPX &gt;()</a>	461
5.15.3.22	<a href="#">is_na&lt; STRSPX &gt;()</a>	461
5.15.3.23	<a href="#">is_nan()</a>	462
5.15.3.24	<a href="#">is_nan&lt; CPLXSPX &gt;()</a>	462
5.15.3.25	<a href="#">is_nan&lt; REALSPX &gt;()</a>	462
5.16	<a href="#">std Namespace Reference</a>	462
5.17	<a href="#">tinyformat Namespace Reference</a>	462
5.17.1	<a href="#">Typedef Documentation</a>	463
5.17.1.1	<a href="#">FormatListRef</a>	463
5.17.2	<a href="#">Function Documentation</a>	463
5.17.2.1	<a href="#">format() [1/2]</a>	463
5.17.2.2	<a href="#">format() [2/2]</a>	464
5.17.2.3	<a href="#">formatValue()</a>	464
5.17.2.4	<a href="#">makeFormatList()</a>	465
5.17.2.5	<a href="#">printf()</a>	465
5.17.2.6	<a href="#">printfln()</a>	466
5.17.2.7	<a href="#">vformat()</a>	466
5.18	<a href="#">tinyformat::detail Namespace Reference</a>	467
5.18.1	<a href="#">Function Documentation</a>	467

---

5.18.1.1	formatImpl()	467
5.18.1.2	formatTruncated()	468
5.18.1.3	parseIntAndAdvance()	469
5.18.1.4	printFormatStringLiteral()	469
5.18.1.5	streamStateFromFormat()	469
<b>6</b>	<b>Class Documentation</b>	<b>471</b>
6.1	Rcpp::traits::__sфинаe_types Struct Reference	471
6.1.1	Detailed Description	472
6.1.2	Member Typedef Documentation	473
6.1.2.1	__one	473
6.2	Rcpp::traits::__sфинаe_types::__two Struct Reference	473
6.2.1	Detailed Description	473
6.2.2	Member Data Documentation	473
6.2.2.1	__arr	473
6.3	Rcpp::traits::__has_iterator_helper< T > Class Template Reference	474
6.3.1	Detailed Description	475
6.3.2	Member Function Documentation	475
6.3.2.1	__test() [1/2]	475
6.3.2.2	__test() [2/2]	475
6.3.3	Member Data Documentation	475
6.3.3.1	value	476
6.4	Rcpp::traits::__has_matrix_interface_helper< T > Class Template Reference	476
6.4.1	Detailed Description	477
6.4.2	Member Function Documentation	477
6.4.2.1	__test() [1/2]	477
6.4.2.2	__test() [2/2]	477
6.4.3	Member Data Documentation	478
6.4.3.1	value	478
6.5	Rcpp::traits::__has_rtype_helper< T > Class Template Reference	478
6.5.1	Detailed Description	479
6.5.2	Member Function Documentation	479
6.5.2.1	__test() [1/2]	480
6.5.2.2	__test() [2/2]	480
6.5.3	Member Data Documentation	480
6.5.3.1	value	480
6.6	Rcpp::traits::__is_eigen_helper< T > Class Template Reference	480
6.6.1	Detailed Description	481
6.6.2	Member Function Documentation	481

---

6.6.2.1	<a href="#">__test() [1/2]</a>	482
6.6.2.2	<a href="#">__test() [2/2]</a>	482
6.6.3	<a href="#">Member Data Documentation</a>	482
6.6.3.1	<a href="#">value</a>	482
6.7	<a href="#">Rcpp::traits::_is_exporter_helper&lt; T &gt; Class Template Reference</a>	482
6.7.1	<a href="#">Detailed Description</a>	483
6.7.2	<a href="#">Member Function Documentation</a>	483
6.7.2.1	<a href="#">__test() [1/2]</a>	484
6.7.2.2	<a href="#">__test() [2/2]</a>	484
6.7.3	<a href="#">Member Data Documentation</a>	484
6.7.3.1	<a href="#">value</a>	484
6.8	<a href="#">Rcpp::traits::_is_generator_helper&lt; T &gt; Class Template Reference</a>	484
6.8.1	<a href="#">Detailed Description</a>	485
6.8.2	<a href="#">Member Function Documentation</a>	485
6.8.2.1	<a href="#">__test() [1/2]</a>	486
6.8.2.2	<a href="#">__test() [2/2]</a>	486
6.8.3	<a href="#">Member Data Documentation</a>	486
6.8.3.1	<a href="#">value</a>	486
6.9	<a href="#">Rcpp::traits::_is_importer_helper&lt; T &gt; Class Template Reference</a>	486
6.9.1	<a href="#">Detailed Description</a>	487
6.9.2	<a href="#">Member Function Documentation</a>	487
6.9.2.1	<a href="#">__test() [1/2]</a>	488
6.9.2.2	<a href="#">__test() [2/2]</a>	488
6.9.3	<a href="#">Member Data Documentation</a>	488
6.9.3.1	<a href="#">value</a>	488
6.10	<a href="#">Rcpp::traits::_is_sugar_expression_helper&lt; T &gt; Class Template Reference</a>	488
6.10.1	<a href="#">Detailed Description</a>	490
6.10.2	<a href="#">Member Function Documentation</a>	490
6.10.2.1	<a href="#">__test() [1/2]</a>	490
6.10.2.2	<a href="#">__test() [2/2]</a>	490
6.10.3	<a href="#">Member Data Documentation</a>	490
6.10.3.1	<a href="#">value</a>	490
6.11	<a href="#">Rcpp::traits::_has_iterator_helper&lt; T &gt;::_Wrap_type&lt; U &gt; Struct Template Reference</a>	491
6.11.1	<a href="#">Detailed Description</a>	491
6.12	<a href="#">Rcpp::traits::_has_matrix_interface_helper&lt; T &gt;::_Wrap_type&lt; U &gt; Struct Template Reference</a>	491
6.12.1	<a href="#">Detailed Description</a>	491
6.13	<a href="#">Rcpp::traits::_has_rtype_helper&lt; T &gt;::_Wrap_type&lt; U &gt; Struct Template Reference</a>	491
6.13.1	<a href="#">Detailed Description</a>	491
6.14	<a href="#">Rcpp::traits::_is_eigen_helper&lt; T &gt;::_Wrap_type&lt; U &gt; Struct Template Reference</a>	492

---

6.14.1 Detailed Description	492
6.15 Rcpp::traits::_is_exporter_helper< T >::_Wrap_type< U > Struct Template Reference	492
6.15.1 Detailed Description	492
6.16 Rcpp::traits::_is_generator_helper< T >::_Wrap_type< U > Struct Template Reference	492
6.16.1 Detailed Description	492
6.17 Rcpp::traits::_is_importer_helper< T >::_Wrap_type< U > Struct Template Reference	493
6.17.1 Detailed Description	493
6.18 Rcpp::traits::_is_sugar_expression_helper< T >::_Wrap_type< U > Struct Template Reference	493
6.18.1 Detailed Description	493
6.19 Rcpp::sugar::All< NA, T > Class Template Reference	493
6.19.1 Detailed Description	494
6.19.2 Member Typedef Documentation	494
6.19.2.1 PARENT	494
6.19.2.2 VEC_TYPE	495
6.19.3 Constructor & Destructor Documentation	495
6.19.3.1 All()	495
6.19.4 Member Function Documentation	495
6.19.4.1 apply()	495
6.19.5 Member Data Documentation	496
6.19.5.1 object	496
6.20 Rcpp::sugar::All< false, T > Class Template Reference	497
6.20.1 Detailed Description	498
6.20.2 Member Typedef Documentation	498
6.20.2.1 PARENT	498
6.20.2.2 VEC_TYPE	498
6.20.3 Constructor & Destructor Documentation	498
6.20.3.1 All()	498
6.20.4 Member Function Documentation	499
6.20.4.1 apply()	499
6.20.5 Member Data Documentation	499
6.20.5.1 object	499
6.21 Rcpp::traits::allowed_matrix_type< bool > Struct Template Reference	500
6.21.1 Detailed Description	500
6.22 Rcpp::traits::allowed_matrix_type< true > Struct Reference	500
6.22.1 Detailed Description	500
6.23 Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	500
6.23.1 Detailed Description	501
6.23.2 Member Typedef Documentation	501

---

6.23.2.1 LHS_TYPE	502
6.23.2.2 RHS_TYPE	502
6.23.3 Constructor & Destructor Documentation	502
6.23.3.1 And_LogicalExpression_LogicalExpression()	502
6.23.4 Member Function Documentation	502
6.23.4.1 operator[]()	502
6.23.4.2 size()	503
6.23.5 Member Data Documentation	503
6.23.5.1 lhs	503
6.23.5.2 rhs	504
6.24 Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > Class Template Reference	504
6.24.1 Detailed Description	505
6.24.2 Member Typedef Documentation	505
6.24.2.1 LHS_TYPE	505
6.24.2.2 RHS_TYPE	505
6.24.3 Constructor & Destructor Documentation	506
6.24.3.1 And_LogicalExpression_LogicalExpression()	506
6.24.4 Member Function Documentation	506
6.24.4.1 operator[]()	506
6.24.4.2 size()	506
6.24.5 Member Data Documentation	507
6.24.5.1 lhs	507
6.24.5.2 rhs	507
6.25 Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > Class Template Reference	507
6.25.1 Detailed Description	508
6.25.2 Member Typedef Documentation	508
6.25.2.1 LHS_TYPE	508
6.25.2.2 RHS_TYPE	508
6.25.3 Constructor & Destructor Documentation	509
6.25.3.1 And_LogicalExpression_LogicalExpression()	509
6.25.4 Member Function Documentation	509
6.25.4.1 operator[]()	509
6.25.4.2 size()	509
6.25.5 Member Data Documentation	510
6.25.5.1 lhs	510
6.25.5.2 rhs	510
6.26 Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > Class Template Reference	510

---

6.26.1 Detailed Description	511
6.26.2 Member Typedef Documentation	511
6.26.2.1 LHS_TYPE	511
6.26.2.2 RHS_TYPE	511
6.26.3 Constructor & Destructor Documentation	512
6.26.3.1 And_LogicalExpression_LogicalExpression()	512
6.26.4 Member Function Documentation	512
6.26.4.1 operator[]()	512
6.26.4.2 size()	512
6.26.5 Member Data Documentation	513
6.26.5.1 lhs	513
6.26.5.2 rhs	513
6.27 Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T > Class Template Reference	513
6.27.1 Detailed Description	514
6.27.2 Member Typedef Documentation	514
6.27.2.1 BASE	515
6.27.2.2 LHS_TYPE	515
6.27.3 Constructor & Destructor Documentation	515
6.27.3.1 And_SingleLogicalResult_bool()	515
6.27.4 Member Function Documentation	515
6.27.4.1 apply()	516
6.27.5 Member Data Documentation	516
6.27.5.1 lhs	516
6.27.5.2 rhs	516
6.28 Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	517
6.28.1 Detailed Description	518
6.28.2 Member Typedef Documentation	518
6.28.2.1 BASE	518
6.28.2.2 LHS_TYPE	518
6.28.2.3 RHS_TYPE	518
6.28.3 Constructor & Destructor Documentation	519
6.28.3.1 And_SingleLogicalResult_SingleLogicalResult()	519
6.28.4 Member Function Documentation	519
6.28.4.1 apply()	519
6.28.5 Member Data Documentation	520
6.28.5.1 lhs	520
6.28.5.2 rhs	520
6.29 Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T > Class Template Reference	521

---

6.29.1 Detailed Description	522
6.29.2 Member Typedef Documentation	522
6.29.2.1 BASE	522
6.29.2.2 LHS_TYPE	522
6.29.2.3 RHS_TYPE	522
6.29.3 Constructor & Destructor Documentation	523
6.29.3.1 And_SingleLogicalResult_SingleLogicalResult()	523
6.29.4 Member Function Documentation	523
6.29.4.1 apply()	523
6.29.5 Member Data Documentation	523
6.29.5.1 lhs	524
6.29.5.2 rhs	524
6.30 Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > Class	
Template Reference	524
6.30.1 Detailed Description	525
6.30.2 Member Typedef Documentation	525
6.30.2.1 BASE	525
6.30.2.2 LHS_TYPE	526
6.30.2.3 RHS_TYPE	526
6.30.3 Constructor & Destructor Documentation	526
6.30.3.1 And_SingleLogicalResult_SingleLogicalResult()	526
6.30.4 Member Function Documentation	526
6.30.4.1 apply()	527
6.30.5 Member Data Documentation	527
6.30.5.1 lhs	527
6.30.5.2 rhs	527
6.31 Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > Class	
Template Reference	528
6.31.1 Detailed Description	529
6.31.2 Member Typedef Documentation	529
6.31.2.1 BASE	529
6.31.2.2 LHS_TYPE	529
6.31.2.3 RHS_TYPE	529
6.31.3 Constructor & Destructor Documentation	530
6.31.3.1 And_SingleLogicalResult_SingleLogicalResult()	530
6.31.4 Member Function Documentation	530
6.31.4.1 apply()	530
6.31.5 Member Data Documentation	530
6.31.5.1 lhs	531
6.31.5.2 rhs	531

---

6.32 Rcpp::sugar::Any< NA, T > Class Template Reference	531
6.32.1 Detailed Description	532
6.32.2 Member Typedef Documentation	532
6.32.2.1 PARENT	532
6.32.2.2 VEC_TYPE	532
6.32.3 Constructor & Destructor Documentation	533
6.32.3.1 Any()	533
6.32.4 Member Function Documentation	533
6.32.4.1 apply()	533
6.32.5 Member Data Documentation	534
6.32.5.1 object	534
6.33 Rcpp::sugar::Any< false, T > Class Template Reference	535
6.33.1 Detailed Description	536
6.33.2 Member Typedef Documentation	536
6.33.2.1 PARENT	536
6.33.2.2 VEC_TYPE	536
6.33.3 Constructor & Destructor Documentation	536
6.33.3.1 Any()	536
6.33.4 Member Function Documentation	537
6.33.4.1 apply()	537
6.33.5 Member Data Documentation	537
6.33.5.1 object	537
6.34 Rcpp::Argument Class Reference	538
6.34.1 Detailed Description	538
6.34.2 Constructor & Destructor Documentation	538
6.34.2.1 Argument() [1/2]	538
6.34.2.2 Argument() [2/2]	538
6.34.3 Member Function Documentation	538
6.34.3.1 operator=()	539
6.34.4 Member Data Documentation	539
6.34.4.1 name	539
6.35 Rcpp::attributes::Argument Class Reference	539
6.35.1 Detailed Description	540
6.35.2 Constructor & Destructor Documentation	540
6.35.2.1 Argument() [1/2]	540
6.35.2.2 Argument() [2/2]	540
6.35.3 Member Function Documentation	541
6.35.3.1 defaultValue()	541
6.35.3.2 empty()	541



---

6.35.3.3 name()	541
6.35.3.4 operator!=(())	542
6.35.3.5 operator==(())	542
6.35.3.6 type()	542
6.35.4 Member Data Documentation	542
6.35.4.1 defaultValue_	542
6.35.4.2 name_	543
6.35.4.3 type_	543
6.36 Rcpp::Armor< T > Class Template Reference	543
6.36.1 Detailed Description	544
6.36.2 Constructor & Destructor Documentation	544
6.36.2.1 Armor() [1/3]	544
6.36.2.2 Armor() [2/3]	545
6.36.2.3 ~Armor()	545
6.36.2.4 Armor() [3/3]	545
6.36.3 Member Function Documentation	545
6.36.3.1 init()	546
6.36.3.2 operator SEXP()	546
6.36.3.3 operator=(()) [1/3]	546
6.36.3.4 operator=(()) [2/3]	546
6.36.3.5 operator=(()) [3/3]	547
6.36.4 Member Data Documentation	547
6.36.4.1 data	547
6.36.4.2 index	547
6.37 Rcpp::attributes::Attribute Class Reference	548
6.37.1 Detailed Description	548
6.37.2 Constructor & Destructor Documentation	548
6.37.2.1 Attribute() [1/2]	548
6.37.2.2 Attribute() [2/2]	549
6.37.3 Member Function Documentation	549
6.37.3.1 customRSignature()	549
6.37.3.2 empty()	550
6.37.3.3 exportedCppName()	550
6.37.3.4 exportedName()	551
6.37.3.5 function()	551
6.37.3.6 hasParameter()	552
6.37.3.7 invisible()	552
6.37.3.8 isExportedFunction()	553
6.37.3.9 name()	553

---

6.37.3.10 operator!=(())	553
6.37.3.11 operator==(())	554
6.37.3.12 paramNamed()	554
6.37.3.13 params()	554
6.37.3.14 rng()	555
6.37.3.15 roxygen()	555
6.37.4 Member Data Documentation	555
6.37.4.1 function_	555
6.37.4.2 name_	556
6.37.4.3 params_	556
6.37.4.4 roxygen_	556
6.38 Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy Class Reference	557
6.38.1 Detailed Description	558
6.38.2 Constructor & Destructor Documentation	558
6.38.2.1 AttributeProxy()	558
6.38.3 Member Function Documentation	558
6.38.3.1 get()	558
6.38.3.2 operator SEXP()	559
6.38.3.3 operator T()	559
6.38.3.4 operator=() [1/3]	559
6.38.3.5 operator=() [2/3]	560
6.38.3.6 operator=() [3/3]	561
6.38.3.7 set()	561
6.38.4 Member Data Documentation	561
6.38.4.1 attr_name	561
6.38.4.2 parent	561
6.39 Rcpp::AttributeProxyPolicy< CLASS > Class Template Reference	562
6.39.1 Detailed Description	562
6.39.2 Member Function Documentation	563
6.39.2.1 attr() [1/2]	563
6.39.2.2 attr() [2/2]	563
6.39.2.3 attributeNames()	563
6.39.2.4 hasAttribute()	564
6.40 Rcpp::stats::BetaGenerator Class Reference	564
6.40.1 Detailed Description	565
6.40.2 Constructor & Destructor Documentation	565
6.40.2.1 BetaGenerator()	565
6.40.3 Member Function Documentation	566
6.40.3.1 operator()()	566

---

6.40.4 Member Data Documentation	566
6.40.4.1 a	566
6.40.4.2 b	566
6.41 Rcpp::traits::is_convertible< T, U >::Big Struct Reference	566
6.41.1 Detailed Description	567
6.41.2 Member Data Documentation	567
6.41.2.1 dummy	567
6.42 Rcpp::binary_call< T1, T2, RESULT_TYPE > Class Template Reference	567
6.42.1 Detailed Description	568
6.42.2 Constructor & Destructor Documentation	568
6.42.2.1 binary_call() [1/3]	569
6.42.2.2 binary_call() [2/3]	569
6.42.2.3 binary_call() [3/3]	569
6.42.3 Member Function Documentation	569
6.42.3.1 operator>()	569
6.42.4 Member Data Documentation	570
6.42.4.1 call	570
6.42.4.2 proxy1	570
6.42.4.3 proxy2	570
6.43 Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E > Class Template Reference	570
6.43.1 Detailed Description	571
6.43.2 Member Typedef Documentation	571
6.43.2.1 stored_type	571
6.43.3 Member Function Documentation	571
6.43.3.1 ncol()	571
6.43.3.2 nrow()	572
6.43.3.3 operator const E &()	572
6.43.3.4 operator E&()	572
6.43.3.5 operator>()	572
6.43.3.6 operator[]()	572
6.43.3.7 size()	573
6.44 Rcpp::BindingPolicy< EnvironmentClass >::Binding Class Reference	573
6.44.1 Detailed Description	574
6.44.2 Constructor & Destructor Documentation	574
6.44.2.1 Binding()	574
6.44.3 Member Function Documentation	575
6.44.3.1 active()	575
6.44.3.2 exists()	575
6.44.3.3 get()	575

---

6.44.3.4 lock()	576
6.44.3.5 locked()	576
6.44.3.6 operator T()	576
6.44.3.7 operator=() [1/3]	577
6.44.3.8 operator=() [2/3]	577
6.44.3.9 operator=() [3/3]	578
6.44.3.10 set()	578
6.44.3.11 unlock()	578
6.44.4 Member Data Documentation	578
6.44.4.1 env	578
6.44.4.2 name	579
6.45 Rcpp::BindingPolicy< EnvironmentClass > Class Template Reference	579
6.45.1 Detailed Description	579
6.45.2 Member Function Documentation	579
6.45.2.1 operator[]() [1/2]	580
6.45.2.2 operator[]() [2/2]	580
6.46 Rcpp::stats::BinomGenerator Class Reference	580
6.46.1 Detailed Description	581
6.46.2 Constructor & Destructor Documentation	581
6.46.2.1 BinomGenerator()	581
6.46.3 Member Function Documentation	582
6.46.3.1 operator>()	582
6.46.4 Member Data Documentation	582
6.46.4.1 nin	582
6.46.4.2 pp	582
6.47 Rcpp::traits::both< T, U > Struct Template Reference	583
6.47.1 Detailed Description	583
6.48 Cache Class Reference	584
6.48.1 Detailed Description	584
6.48.2 Member Typedef Documentation	584
6.48.2.1 iterator [1/2]	584
6.48.2.2 iterator [2/2]	585
6.48.2.3 proxy [1/2]	585
6.48.2.4 proxy [2/2]	585
6.48.3 Constructor & Destructor Documentation	585
6.48.3.1 Cache() [1/2]	585
6.48.3.2 Cache() [2/2]	585
6.48.4 Member Function Documentation	586
6.48.4.1 ref() [1/4]	586

---

6.48.4.2 ref() [2/4] . . . . .	586
6.48.4.3 ref() [3/4] . . . . .	586
6.48.4.4 ref() [4/4] . . . . .	586
6.48.5 Member Data Documentation . . . . .	587
6.48.5.1 data . . . . .	587
6.49 Rcpp::can_have_na< T > Struct Template Reference . . . . .	587
6.49.1 Detailed Description . . . . .	588
6.50 Rcpp::Matrix< RTYPE, StoragePolicy >::can_have_na Struct Reference . . . . .	588
6.50.1 Detailed Description . . . . .	589
6.51 Rcpp::MatrixBase< RTYPE, na, MATRIX >::can_have_na Struct Reference . . . . .	589
6.51.1 Detailed Description . . . . .	590
6.52 Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na Struct Reference . . . . .	590
6.52.1 Detailed Description . . . . .	591
6.53 Rcpp::stats::CauchyGenerator Class Reference . . . . .	592
6.53.1 Detailed Description . . . . .	593
6.53.2 Constructor & Destructor Documentation . . . . .	593
6.53.2.1 CauchyGenerator() . . . . .	593
6.53.3 Member Function Documentation . . . . .	593
6.53.3.1 operator>() . . . . .	593
6.53.4 Member Data Documentation . . . . .	594
6.53.4.1 location . . . . .	594
6.53.4.2 scale . . . . .	594
6.54 Rcpp::stats::CauchyGenerator_0 Class Reference . . . . .	594
6.54.1 Detailed Description . . . . .	595
6.54.2 Constructor & Destructor Documentation . . . . .	595
6.54.2.1 CauchyGenerator_0() . . . . .	595
6.54.3 Member Function Documentation . . . . .	595
6.54.3.1 operator>() . . . . .	596
6.55 Rcpp::stats::CauchyGenerator_1 Class Reference . . . . .	596
6.55.1 Detailed Description . . . . .	597
6.55.2 Constructor & Destructor Documentation . . . . .	597
6.55.2.1 CauchyGenerator_1() . . . . .	597
6.55.3 Member Function Documentation . . . . .	598
6.55.3.1 operator>() . . . . .	598
6.55.4 Member Data Documentation . . . . .	598
6.55.4.1 location . . . . .	598
6.56 Rcpp::sugar::cbind_impl::cbind_sexptype_traits< T > Struct Template Reference . . . . .	599
6.56.1 Detailed Description . . . . .	599
6.57 Rcpp::sugar::cbind_impl::cbind_sexptype_traits< SEXP > Struct Reference . . . . .	600

---

6.57.1 Detailed Description	600
6.57.2 Member Enumeration Documentation	600
6.57.2.1 anonymous enum	600
6.58 Rcpp::sugar::cbind_impl::cbind_storage_type< RTYPE > Struct Template Reference	601
6.58.1 Detailed Description	601
6.59 Rcpp::sugar::cbind_impl::cbind_storage_type< LGLSXP > Struct Reference	602
6.59.1 Detailed Description	602
6.59.2 Member Typedef Documentation	602
6.59.2.1 type	602
6.60 Rcpp::traits::char_type< T > Struct Template Reference	602
6.60.1 Detailed Description	603
6.60.2 Member Typedef Documentation	603
6.60.2.1 type	603
6.61 Rcpp::traits::char_type< const char * > Struct Reference	603
6.61.1 Detailed Description	603
6.61.2 Member Typedef Documentation	603
6.61.2.1 type	604
6.62 Rcpp::traits::char_type< const wchar_t * > Struct Reference	604
6.62.1 Detailed Description	604
6.62.2 Member Typedef Documentation	604
6.62.2.1 type	604
6.63 Rcpp::ChildVector< T > Class Template Reference	605
6.63.1 Detailed Description	606
6.63.2 Constructor & Destructor Documentation	606
6.63.2.1 ChildVector() [1/2]	606
6.63.2.2 ChildVector() [2/2]	606
6.63.3 Member Function Documentation	606
6.63.3.1 operator=() [1/3]	606
6.63.3.2 operator=() [2/3]	607
6.63.3.3 operator=() [3/3]	607
6.63.4 Member Data Documentation	607
6.63.4.1 i	607
6.63.4.2 parent	608
6.64 Rcpp::stats::ChisqGenerator Class Reference	608
6.64.1 Detailed Description	609
6.64.2 Constructor & Destructor Documentation	609
6.64.2.1 ChisqGenerator()	609
6.64.3 Member Function Documentation	609
6.64.3.1 operator>()	609

---

6.64.4 Member Data Documentation	609
6.64.4.1 df_2	610
6.65 Rcpp::sugar::clamp_operator< RTYPE, NA > Struct Template Reference	610
6.65.1 Detailed Description	610
6.65.2 Member Typedef Documentation	610
6.65.2.1 STORAGE	611
6.65.3 Constructor & Destructor Documentation	611
6.65.3.1 clamp_operator()	611
6.65.4 Member Function Documentation	611
6.65.4.1 operator>()	611
6.65.5 Member Data Documentation	611
6.65.5.1 lhs	612
6.65.5.2 rhs	612
6.66 Rcpp::sugar::clamp_operator< REALSXP, true > Struct Reference	612
6.66.1 Detailed Description	612
6.66.2 Constructor & Destructor Documentation	613
6.66.2.1 clamp_operator()	613
6.66.3 Member Function Documentation	613
6.66.3.1 operator>()	613
6.66.4 Member Data Documentation	613
6.66.4.1 lhs	614
6.66.4.2 rhs	614
6.67 Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T > Class Template Reference	614
6.67.1 Detailed Description	615
6.67.2 Member Typedef Documentation	616
6.67.2.1 OPERATOR	616
6.67.2.2 STORAGE	616
6.67.3 Constructor & Destructor Documentation	616
6.67.3.1 Clamp_Primitive_Vector_Primitive()	616
6.67.4 Member Function Documentation	616
6.67.4.1 operator[]()	617
6.67.4.2 size()	617
6.67.5 Member Data Documentation	617
6.67.5.1 op	617
6.67.5.2 vec	617
6.68 class_< Class > Class Template Reference	618
6.68.1 Detailed Description	620
6.68.2 Member Typedef Documentation	620
6.68.2.1 constructor_class	620

---

6.68.2.2	factory_class	620
6.68.2.3	finalizer_class	621
6.68.2.4	map_vec_signed_method	621
6.68.2.5	method_class	621
6.68.2.6	METHOD_MAP	621
6.68.2.7	PAIR	621
6.68.2.8	prop_class	622
6.68.2.9	PROP_PAIR	622
6.68.2.10	PROPERTY_MAP	622
6.68.2.11	self	622
6.68.2.12	signed_constructor_class	622
6.68.2.13	signed_factory_class	623
6.68.2.14	signed_method_class	623
6.68.2.15	vec_signed_constructor	623
6.68.2.16	vec_signed_factory	623
6.68.2.17	vec_signed_method	623
6.68.2.18	vec_signed_method_pair	624
6.68.2.19	XP	624
6.68.3	Constructor & Destructor Documentation	624
6.68.3.1	class_() [1/2]	624
6.68.3.2	~class_()	625
6.68.3.3	class_() [2/2]	625
6.68.4	Member Function Documentation	625
6.68.4.1	AddConstructor()	625
6.68.4.2	AddFactory()	625
6.68.4.3	AddMethod()	626
6.68.4.4	AddProperty()	626
6.68.4.5	complete()	627
6.68.4.6	default_constructor()	627
6.68.4.7	derives()	627
6.68.4.8	fields()	628
6.68.4.9	finalizer()	628
6.68.4.10	get_instance()	629
6.68.4.11	get_typeinfo_name()	629
6.68.4.12	getConstructors()	629
6.68.4.13	getMethods()	630
6.68.4.14	getProperty()	630
6.68.4.15	has_default_constructor()	630
6.68.4.16	has_method()	631



---

6.68.4.17	has_property()	631
6.68.4.18	invoke()	631
6.68.4.19	invoke_notvoid()	632
6.68.4.20	invoke_void()	632
6.68.4.21	method_names()	632
6.68.4.22	methods_arity()	633
6.68.4.23	methods_voidness()	633
6.68.4.24	newInstance()	634
6.68.4.25	property_class()	634
6.68.4.26	property_classes()	634
6.68.4.27	property_is_readonly()	635
6.68.4.28	property_names()	635
6.68.4.29	run_finalizer()	635
6.68.4.30	SetFinalizer()	635
6.68.4.31	setProperty()	636
6.68.5	Member Data Documentation	636
6.68.5.1	class_pointer	636
6.68.5.2	constructors	636
6.68.5.3	factories	636
6.68.5.4	finalizer_pointer	637
6.68.5.5	properties	637
6.68.5.6	specials	637
6.68.5.7	typeid_name	637
6.68.5.8	vec_methods	638
6.69	Rcpp::class_Base Class Reference	638
6.69.1	Detailed Description	639
6.69.2	Member Typedef Documentation	639
6.69.2.1	ENUM	639
6.69.2.2	ENUM_MAP	639
6.69.2.3	ENUM_MAP_PAIR	639
6.69.2.4	XP_Class	639
6.69.3	Constructor & Destructor Documentation	640
6.69.3.1	class_Base() [1/2]	640
6.69.3.2	class_Base() [2/2]	640
6.69.3.3	~class_Base()	640
6.69.4	Member Function Documentation	640
6.69.4.1	add_enum()	640
6.69.4.2	complete()	641
6.69.4.3	fields()	641

---

6.69.4.4	<a href="#">get_typeinfo_name()</a>	641
6.69.4.5	<a href="#">getConstructors()</a>	641
6.69.4.6	<a href="#">getMethods()</a>	641
6.69.4.7	<a href="#">getProperty()</a>	642
6.69.4.8	<a href="#">has_default_constructor()</a>	642
6.69.4.9	<a href="#">has_method()</a>	642
6.69.4.10	<a href="#">has_property()</a>	642
6.69.4.11	<a href="#">has_typeinfo_name()</a>	642
6.69.4.12	<a href="#">invoke()</a>	643
6.69.4.13	<a href="#">invoke_notvoid()</a>	643
6.69.4.14	<a href="#">invoke_void()</a>	643
6.69.4.15	<a href="#">method_names()</a>	643
6.69.4.16	<a href="#">methods_arity()</a>	644
6.69.4.17	<a href="#">methods_voidness()</a>	644
6.69.4.18	<a href="#">newInstance()</a>	644
6.69.4.19	<a href="#">property_class()</a>	644
6.69.4.20	<a href="#">property_classes()</a>	644
6.69.4.21	<a href="#">property_is_readonly()</a>	645
6.69.4.22	<a href="#">property_names()</a>	645
6.69.4.23	<a href="#">run_finalizer()</a>	645
6.69.4.24	<a href="#">setProperty()</a>	645
6.69.5	<a href="#">Member Data Documentation</a>	645
6.69.5.1	<a href="#">docstring</a>	645
6.69.5.2	<a href="#">enums</a>	646
6.69.5.3	<a href="#">name</a>	646
6.69.5.4	<a href="#">parents</a>	646
6.70	<a href="#">Rcpp::sugar::Col&lt; RTYPE, LHS_NA, LHS_T &gt; Class Template Reference</a>	647
6.70.1	<a href="#">Detailed Description</a>	648
6.70.2	<a href="#">Member Typedef Documentation</a>	649
6.70.2.1	<a href="#">LHS_TYPE</a>	649
6.70.3	<a href="#">Constructor &amp; Destructor Documentation</a>	649
6.70.3.1	<a href="#">Col()</a>	649
6.70.4	<a href="#">Member Function Documentation</a>	649
6.70.4.1	<a href="#">ncol()</a>	649
6.70.4.2	<a href="#">nrow()</a>	650
6.70.4.3	<a href="#">operator&gt;()()</a>	650
6.70.4.4	<a href="#">size()</a>	650
6.70.5	<a href="#">Member Data Documentation</a>	650
6.70.5.1	<a href="#">nc</a>	650

---

6.70.5.2 nr	651
6.71 Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM > Class Template Reference	651
6.71.1 Detailed Description	652
6.71.2 Member Typedef Documentation	652
6.71.2.1 return_traits	652
6.71.2.2 return_vector	652
6.71.2.3 stored_type	653
6.71.3 Constructor & Destructor Documentation	653
6.71.3.1 ColMeansImpl()	653
6.71.4 Member Function Documentation	653
6.71.4.1 get()	653
6.71.5 Member Data Documentation	654
6.71.5.1 ref	654
6.72 Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA_RM > Class Template Reference	654
6.72.1 Detailed Description	655
6.73 Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true > Class Template Reference	655
6.73.1 Detailed Description	656
6.73.2 Member Typedef Documentation	657
6.73.2.1 return_traits	657
6.73.2.2 return_vector	657
6.73.2.3 stored_type	657
6.73.3 Constructor & Destructor Documentation	657
6.73.3.1 ColMeansImpl()	657
6.73.4 Member Function Documentation	658
6.73.4.1 get()	658
6.73.5 Member Data Documentation	658
6.73.5.1 ref	659
6.74 Rcpp::sugar::detail::ColMeansReturn< RTYPE > Struct Template Reference	659
6.74.1 Detailed Description	660
6.75 Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM > Class Template Reference	660
6.75.1 Detailed Description	661
6.75.2 Member Typedef Documentation	661
6.75.2.1 return_traits	661
6.75.2.2 return_vector	661
6.75.2.3 stored_type	662
6.75.3 Constructor & Destructor Documentation	662
6.75.3.1 ColSumsImpl()	662
6.75.4 Member Function Documentation	662
6.75.4.1 get()	662

---

6.75.5 Member Data Documentation	663
6.75.5.1 ref	663
6.76 Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA_RM > Class Template Reference	663
6.76.1 Detailed Description	664
6.77 Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true > Class Template Reference	664
6.77.1 Detailed Description	665
6.77.2 Member Typedef Documentation	666
6.77.2.1 return_traits	666
6.77.2.2 return_vector	666
6.77.2.3 stored_type	666
6.77.3 Constructor & Destructor Documentation	666
6.77.3.1 ColSumsImpl()	666
6.77.4 Member Function Documentation	667
6.77.4.1 get()	667
6.77.5 Member Data Documentation	667
6.77.5.1 ref	667
6.78 Rcpp::sugar::detail::ColSumsReturn< RTYPE > Struct Template Reference	668
6.78.1 Detailed Description	668
6.79 Rcpp::attributes::CommentState Class Reference	669
6.79.1 Detailed Description	669
6.79.2 Constructor & Destructor Documentation	669
6.79.2.1 CommentState() [1/2]	669
6.79.2.2 CommentState() [2/2]	669
6.79.3 Member Function Documentation	670
6.79.3.1 inComment()	670
6.79.3.2 operator=()	670
6.79.3.3 reset()	670
6.79.3.4 submitLine()	670
6.79.4 Member Data Documentation	670
6.79.4.1 inComment_	671
6.80 Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	671
6.80.1 Detailed Description	672
6.80.2 Member Typedef Documentation	672
6.80.2.1 LHS_TYPE	672
6.80.2.2 RHS_TYPE	672
6.80.2.3 STORAGE	673
6.80.3 Constructor & Destructor Documentation	673
6.80.3.1 Comparator()	673

---

6.80.4 Member Function Documentation	673
6.80.4.1 operator[]()	673
6.80.4.2 size()	674
6.80.5 Member Data Documentation	674
6.80.5.1 lhs	674
6.80.5.2 op	675
6.80.5.3 rhs	675
6.81 Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T > Class Template Reference	675
6.81.1 Detailed Description	676
6.81.2 Member Typedef Documentation	676
6.81.2.1 LHS_TYPE	677
6.81.2.2 RHS_TYPE	677
6.81.2.3 STORAGE	677
6.81.3 Constructor & Destructor Documentation	677
6.81.3.1 Comparator()	677
6.81.4 Member Function Documentation	677
6.81.4.1 operator[]()	678
6.81.4.2 size()	678
6.81.5 Member Data Documentation	678
6.81.5.1 lhs	678
6.81.5.2 op	679
6.81.5.3 rhs	679
6.82 Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T > Class Template Reference	679
6.82.1 Detailed Description	680
6.82.2 Member Typedef Documentation	680
6.82.2.1 LHS_TYPE	680
6.82.2.2 RHS_TYPE	681
6.82.2.3 STORAGE	681
6.82.3 Constructor & Destructor Documentation	681
6.82.3.1 Comparator()	681
6.82.4 Member Function Documentation	681
6.82.4.1 operator[]()	681
6.82.4.2 size()	682
6.82.5 Member Data Documentation	682
6.82.5.1 lhs	682
6.82.5.2 op	682
6.82.5.3 rhs	683
6.83 Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T > Class Template Reference	683

---

6.83.1 Detailed Description	684
6.83.2 Member Typedef Documentation	684
6.83.2.1 METHOD	684
6.83.2.2 STORAGE	685
6.83.2.3 VEC_TYPE	685
6.83.3 Constructor & Destructor Documentation	685
6.83.3.1 Comparator_With_One_Value()	685
6.83.4 Member Function Documentation	686
6.83.4.1 operator[]()	686
6.83.4.2 rhs_is_na()	686
6.83.4.3 rhs_is_not_na()	686
6.83.4.4 size()	687
6.83.5 Member Data Documentation	687
6.83.5.1 lhs	687
6.83.5.2 m	687
6.83.5.3 op	688
6.83.5.4 rhs	688
6.84 Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T > Class Template Reference	688
6.84.1 Detailed Description	689
6.84.2 Member Typedef Documentation	690
6.84.2.1 METHOD	690
6.84.2.2 STORAGE	690
6.84.2.3 VEC_TYPE	690
6.84.3 Constructor & Destructor Documentation	690
6.84.3.1 Comparator_With_One_Value()	691
6.84.4 Member Function Documentation	691
6.84.4.1 operator[]()	691
6.84.4.2 rhs_is_na()	692
6.84.4.3 rhs_is_not_na()	692
6.84.4.4 size()	692
6.84.5 Member Data Documentation	693
6.84.5.1 lhs	693
6.84.5.2 m	693
6.84.5.3 op	693
6.84.5.4 rhs	693
6.85 Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy Class Reference	694
6.85.1 Detailed Description	695
6.85.2 Constructor & Destructor Documentation	695
6.85.2.1 const_AttributeProxy()	695

---

6.85.3 Member Function Documentation	695
6.85.3.1 get()	695
6.85.3.2 operator SEXP()	696
6.85.3.3 operator T()	696
6.85.4 Member Data Documentation	696
6.85.4.1 attr_name	697
6.85.4.2 parent	697
6.86 Rcpp::BindingPolicy< EnvironmentClass >::const_Binding Class Reference	697
6.86.1 Detailed Description	698
6.86.2 Constructor & Destructor Documentation	699
6.86.2.1 const_Binding()	699
6.86.3 Member Function Documentation	699
6.86.3.1 active()	699
6.86.3.2 exists()	699
6.86.3.3 get()	700
6.86.3.4 locked()	700
6.86.3.5 operator T()	700
6.86.4 Member Data Documentation	700
6.86.4.1 env	701
6.86.4.2 name	701
6.87 Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy Class Reference	701
6.87.1 Detailed Description	702
6.87.2 Constructor & Destructor Documentation	702
6.87.2.1 const_DottedPairProxy()	703
6.87.3 Member Function Documentation	703
6.87.3.1 get()	703
6.87.3.2 operator SEXP()	703
6.87.3.3 operator T()	704
6.87.4 Member Data Documentation	704
6.87.4.1 node	704
6.88 Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy Class Reference	705
6.88.1 Detailed Description	706
6.88.2 Constructor & Destructor Documentation	706
6.88.2.1 const_FieldProxy()	706
6.88.3 Member Function Documentation	706
6.88.3.1 get()	707
6.88.3.2 operator SEXP()	707
6.88.3.3 operator T()	708
6.88.4 Member Data Documentation	708

---

6.88.4.1 field_name	708
6.88.4.2 parent	708
6.89 Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy > Class Template Reference	709
6.89.1 Detailed Description	710
6.89.2 Member Typedef Documentation	710
6.89.2.1 VECTOR	710
6.89.3 Constructor & Destructor Documentation	710
6.89.3.1 const_generic_proxy() [1/3]	711
6.89.3.2 const_generic_proxy() [2/3]	711
6.89.3.3 const_generic_proxy() [3/3]	711
6.89.4 Member Function Documentation	711
6.89.4.1 get()	711
6.89.4.2 move()	712
6.89.4.3 operator bool()	712
6.89.4.4 operator int()	712
6.89.4.5 operator SEXP()	713
6.89.4.6 operator U()	713
6.89.5 Member Data Documentation	713
6.89.5.1 index	714
6.89.5.2 parent	714
6.90 Rcpp::MatrixRow< RTYPE >::const_iter_traits Struct Reference	714
6.90.1 Detailed Description	714
6.90.2 Member Typedef Documentation	715
6.90.2.1 difference_type	715
6.90.2.2 pointer	715
6.90.2.3 reference	715
6.90.2.4 value_type	715
6.90.2.5 vector_iterator	716
6.91 Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits Struct Reference	716
6.91.1 Detailed Description	716
6.91.2 Member Typedef Documentation	716
6.91.2.1 difference_type	716
6.91.2.2 iterator_category	717
6.91.2.3 pointer	717
6.91.2.4 reference	717
6.91.2.5 value_type	717
6.92 Rcpp::ConstMatrixRow< RTYPE >::const_iterator Class Reference	718
6.92.1 Detailed Description	719
6.92.2 Member Typedef Documentation	719



---

6.92.2.1 difference_type	719
6.92.2.2 iterator_category	719
6.92.2.3 pointer	719
6.92.2.4 reference	720
6.92.2.5 value_type	720
6.92.2.6 vector_iterator	720
6.92.3 Constructor & Destructor Documentation	720
6.92.3.1 const_iterator() [1/2]	720
6.92.3.2 const_iterator() [2/2]	721
6.92.4 Member Function Documentation	721
6.92.4.1 operator!=(())	721
6.92.4.2 operator*(())	721
6.92.4.3 operator+()	721
6.92.4.4 operator++() [1/2]	722
6.92.4.5 operator++() [2/2]	722
6.92.4.6 operator+=(())	722
6.92.4.7 operator-() [1/3]	722
6.92.4.8 operator-() [2/3]	723
6.92.4.9 operator-() [3/3]	723
6.92.4.10 operator--() [1/2]	723
6.92.4.11 operator--() [2/2]	723
6.92.4.12 operator-=(())	724
6.92.4.13 operator->()	724
6.92.4.14 operator<()	724
6.92.4.15 operator<=(())	724
6.92.4.16 operator==(())	725
6.92.4.17 operator>()	725
6.92.4.18 operator>=(())	725
6.92.4.19 operator[]()	725
6.92.5 Member Data Documentation	726
6.92.5.1 index	726
6.92.5.2 row	726
6.93 Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy Class Reference	727
6.93.1 Detailed Description	728
6.93.2 Constructor & Destructor Documentation	728
6.93.2.1 const_NamesProxy()	728
6.93.3 Member Function Documentation	728
6.93.3.1 get()	728
6.93.3.2 operator T()	729

---

6.93.4 Member Data Documentation	729
6.93.4.1 parent	729
6.94 Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy Class Reference	730
6.94.1 Detailed Description	731
6.94.2 Constructor & Destructor Documentation	731
6.94.2.1 const_ProtectedProxy()	731
6.94.3 Member Function Documentation	731
6.94.3.1 get()	731
6.94.3.2 operator SEXP()	732
6.94.3.3 operator U()	732
6.94.4 Member Data Documentation	732
6.94.4.1 xp	733
6.95 Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy Class Reference	733
6.95.1 Detailed Description	734
6.95.2 Constructor & Destructor Documentation	734
6.95.2.1 const_SlotProxy()	734
6.95.3 Member Function Documentation	734
6.95.3.1 get()	735
6.95.3.2 operator SEXP()	735
6.95.3.3 operator T()	735
6.95.4 Member Data Documentation	736
6.95.4.1 parent	736
6.95.4.2 slot_name	736
6.96 Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy > Class Template Reference	737
6.96.1 Detailed Description	738
6.96.2 Member Typedef Documentation	738
6.96.2.1 iterator	738
6.96.2.2 reference	738
6.96.2.3 VECTOR	739
6.96.3 Constructor & Destructor Documentation	739
6.96.3.1 const_string_proxy() [1/4]	739
6.96.3.2 const_string_proxy() [2/4]	739
6.96.3.3 const_string_proxy() [3/4]	739
6.96.3.4 const_string_proxy() [4/4]	740
6.96.4 Member Function Documentation	740
6.96.4.1 begin()	740
6.96.4.2 empty()	741
6.96.4.3 end()	741
6.96.4.4 get()	742

---

6.96.4.5 import()	742
6.96.4.6 move()	742
6.96.4.7 operator char *()	743
6.96.4.8 operator SEXP()	743
6.96.4.9 operator"!=( ) [1/3]	744
6.96.4.10 operator"!=( ) [2/3]	744
6.96.4.11 operator"!=( ) [3/3]	745
6.96.4.12 operator==( ) [1/3]	745
6.96.4.13 operator==( ) [2/3]	746
6.96.4.14 operator==( ) [3/3]	746
6.96.4.15 operator[]()	747
6.96.4.16 size()	747
6.96.5 Friends And Related Function Documentation	748
6.96.5.1 operator+	748
6.96.5.2 operator<<	748
6.96.6 Member Data Documentation	748
6.96.6.1 buffer	748
6.96.6.2 index	748
6.96.6.3 parent	749
6.97 Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy Class Reference	749
6.97.1 Detailed Description	750
6.97.2 Constructor & Destructor Documentation	750
6.97.2.1 const_TagProxy()	751
6.97.3 Member Function Documentation	751
6.97.3.1 get()	751
6.97.3.2 operator SEXP()	751
6.97.3.3 operator U()	752
6.97.4 Member Data Documentation	752
6.97.4.1 xp	752
6.98 Rcpp::ConstInputParameter< T > Class Template Reference	752
6.98.1 Detailed Description	753
6.98.2 Member Typedef Documentation	753
6.98.2.1 const_nonref	753
6.98.3 Constructor & Destructor Documentation	753
6.98.3.1 ConstInputParameter()	753
6.98.4 Member Function Documentation	753
6.98.4.1 operator const_nonref()	753
6.98.5 Member Data Documentation	754
6.98.5.1 obj	754

---

6.99 Rcpp::ConstMatrixColumn< RTYPE > Class Template Reference	754
6.99.1 Detailed Description	756
6.99.2 Member Typedef Documentation	756
6.99.2.1 const_iterator	756
6.99.2.2 const_Proxy	756
6.99.2.3 MATRIX	756
6.99.2.4 value_type	756
6.99.3 Constructor & Destructor Documentation	757
6.99.3.1 ConstMatrixColumn() [1/2]	757
6.99.3.2 ConstMatrixColumn() [2/2]	757
6.99.4 Member Function Documentation	757
6.99.4.1 begin()	757
6.99.4.2 cbegin()	758
6.99.4.3 cend()	758
6.99.4.4 end()	758
6.99.4.5 operator[]()	758
6.99.4.6 size()	759
6.99.5 Member Data Documentation	759
6.99.5.1 const_start	759
6.99.5.2 n	759
6.100 Rcpp::ConstMatrixRow< RTYPE > Class Template Reference	760
6.100.1 Detailed Description	761
6.100.2 Member Typedef Documentation	761
6.100.2.1 const_reference	761
6.100.2.2 iterator	762
6.100.2.3 MATRIX	762
6.100.2.4 value_type	762
6.100.3 Constructor & Destructor Documentation	762
6.100.3.1 ConstMatrixRow() [1/2]	762
6.100.3.2 ConstMatrixRow() [2/2]	763
6.100.4 Member Function Documentation	763
6.100.4.1 begin()	763
6.100.4.2 end()	763
6.100.4.3 get_parent_index()	764
6.100.4.4 operator[]()	764
6.100.4.5 size()	764
6.100.5 Member Data Documentation	765
6.100.5.1 parent	765
6.100.5.2 parent_nrow	765

6.100.5.3 row	765
6.100.5.4 start	765
6.101 Rcpp::ConstReferenceInputParameter< T > Class Template Reference	766
6.101.1 Detailed Description	766
6.101.2 Member Typedef Documentation	766
6.101.2.1 const_reference	766
6.101.3 Constructor & Destructor Documentation	766
6.101.3.1 ConstReferenceInputParameter()	767
6.101.4 Member Function Documentation	767
6.101.4.1 operator const_reference()	767
6.101.5 Member Data Documentation	767
6.101.5.1 obj	767
6.102 Rcpp::traits::container_exporter< Container, double > Struct Template Reference	767
6.102.1 Detailed Description	768
6.102.2 Member Typedef Documentation	768
6.102.2.1 type	768
6.103 Rcpp::traits::container_exporter< Container, int > Struct Template Reference	768
6.103.1 Detailed Description	768
6.103.2 Member Typedef Documentation	769
6.103.2.1 type	769
6.104 Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T > Class Template Reference	769
6.104.1 Detailed Description	770
6.104.2 Member Typedef Documentation	770
6.104.2.1 stored_type	770
6.104.3 Constructor & Destructor Documentation	771
6.104.3.1 ContainerBindable() [1/3]	771
6.104.3.2 ContainerBindable() [2/3]	771
6.104.3.3 ContainerBindable() [3/3]	771
6.104.4 Member Function Documentation	771
6.104.4.1 ncol()	771
6.104.4.2 nrow()	772
6.104.4.3 operator>()	772
6.104.4.4 operator[]()	772
6.104.4.5 size()	772
6.104.5 Member Data Documentation	773
6.104.5.1 len	773
6.104.5.2 nc	773
6.104.5.3 nr	773
6.104.5.4 vec	773

---

6.105 Rcpp::traits::ContainerExporter< ContainerTemplate, T > Class Template Reference . . . . .	774
6.105.1 Detailed Description . . . . .	774
6.105.2 Member Typedef Documentation . . . . .	774
6.105.2.1 Container . . . . .	774
6.105.3 Constructor & Destructor Documentation . . . . .	775
6.105.3.1 ContainerExporter() . . . . .	775
6.105.3.2 ~ContainerExporter() . . . . .	775
6.105.4 Member Function Documentation . . . . .	775
6.105.4.1 get() . . . . .	775
6.105.5 Member Data Documentation . . . . .	775
6.105.5.1 object . . . . .	775
6.105.5.2 RTYPE . . . . .	776
6.106 Rcpp::sugar::conversion_to_bool_is_forbidden< x > Class Template Reference . . . . .	776
6.106.1 Detailed Description . . . . .	777
6.106.2 Member Function Documentation . . . . .	777
6.106.2.1 touch() . . . . .	777
6.107 tinyformat::detail::convertToInt< T, convertible > Struct Template Reference . . . . .	778
6.107.1 Detailed Description . . . . .	778
6.107.2 Member Function Documentation . . . . .	778
6.107.2.1 invoke() . . . . .	778
6.108 tinyformat::detail::convertToInt< T, true > Struct Template Reference . . . . .	778
6.108.1 Detailed Description . . . . .	779
6.108.2 Member Function Documentation . . . . .	779
6.108.2.1 invoke() . . . . .	779
6.109 Rcpp::sugar::CountInserter< HASH, STORAGE > Class Template Reference . . . . .	779
6.109.1 Detailed Description . . . . .	780
6.109.2 Constructor & Destructor Documentation . . . . .	780
6.109.2.1 CountInserter() . . . . .	780
6.109.3 Member Function Documentation . . . . .	780
6.109.3.1 operator>() . . . . .	780
6.109.4 Member Data Documentation . . . . .	780
6.109.4.1 hash . . . . .	781
6.110 Rcpp::CppClass Class Reference . . . . .	781
6.110.1 Detailed Description . . . . .	782
6.110.2 Member Typedef Documentation . . . . .	782
6.110.2.1 Base . . . . .	782
6.110.2.2 XP . . . . .	782
6.110.2.3 XP_Class . . . . .	782
6.110.3 Constructor & Destructor Documentation . . . . .	783

---

6.110.3.1 CppClass() [1/2]	783
6.110.3.2 CppClass() [2/2]	783
6.111 Rcpp::attributes::CppExportsGenerator Class Reference	784
6.111.1 Detailed Description	785
6.111.2 Constructor & Destructor Documentation	785
6.111.2.1 CppExportsGenerator()	785
6.111.3 Member Function Documentation	785
6.111.3.1 commit()	785
6.111.3.2 doWriteFunctions()	786
6.111.3.3 registerCCallable()	787
6.111.3.4 writeBegin()	788
6.111.3.5 writeEnd()	788
6.111.4 Member Data Documentation	789
6.111.4.1 cppExports_	789
6.111.4.2 initFunctions_	790
6.111.4.3 modules_	790
6.111.4.4 nativeRoutines_	790
6.112 Rcpp::attributes::CppExportsIncludeGenerator Class Reference	791
6.112.1 Detailed Description	792
6.112.2 Constructor & Destructor Documentation	792
6.112.2.1 CppExportsIncludeGenerator()	792
6.112.3 Member Function Documentation	792
6.112.3.1 commit()	792
6.112.3.2 doWriteFunctions()	793
6.112.3.3 getCCallable()	794
6.112.3.4 getHeaderGuard()	795
6.112.3.5 writeBegin()	796
6.112.3.6 writeEnd()	796
6.112.4 Member Data Documentation	797
6.112.4.1 includeDir_	797
6.113 Rcpp::CppFinalizer< Class > Class Template Reference	797
6.113.1 Detailed Description	798
6.113.2 Constructor & Destructor Documentation	798
6.113.2.1 CppFinalizer()	798
6.113.3 Member Function Documentation	798
6.113.3.1 run()	798
6.114 Rcpp::CppFunction Class Reference	799
6.114.1 Detailed Description	800
6.114.2 Constructor & Destructor Documentation	800

---

6.114.2.1 CppFunction()	800
6.114.2.2 ~CppFunction()	800
6.114.3 Member Function Documentation	800
6.114.3.1 get_formals()	800
6.114.3.2 get_function_ptr()	801
6.114.3.3 is_void()	801
6.114.3.4 nargs()	801
6.114.3.5 signature()	801
6.114.4 Member Data Documentation	802
6.114.4.1 docstring	802
6.115 Rcpp::CppFunctionBase Class Reference	802
6.115.1 Detailed Description	803
6.115.2 Constructor & Destructor Documentation	803
6.115.2.1 CppFunctionBase()	803
6.115.2.2 ~CppFunctionBase()	803
6.115.3 Member Function Documentation	803
6.115.3.1 operator>()	803
6.116 Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args > Class Template Reference	804
6.116.1 Detailed Description	805
6.116.2 Constructor & Destructor Documentation	805
6.116.2.1 CppFunctionBaseFromStdFunction()	805
6.116.2.2 ~CppFunctionBaseFromStdFunction()	805
6.116.3 Member Function Documentation	805
6.116.3.1 operator>()	806
6.116.4 Member Data Documentation	806
6.116.4.1 fun	806
6.117 Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... > Class Template Reference	807
6.117.1 Detailed Description	808
6.117.2 Constructor & Destructor Documentation	808
6.117.2.1 CppFunctionBaseFromStdFunction()	808
6.117.2.2 ~CppFunctionBaseFromStdFunction()	808
6.117.3 Member Function Documentation	808
6.117.3.1 operator>()	809
6.117.4 Member Data Documentation	809
6.117.4.1 fun	809
6.118 Rcpp::CppInheritedMethod< Class, Parent > Class Template Reference	809
6.118.1 Detailed Description	811
6.118.2 Member Typedef Documentation	811



---

6.118.2.1 ParentMethod	811
6.118.2.2 XP	811
6.118.3 Constructor & Destructor Documentation	811
6.118.3.1 CppInheritedMethod()	811
6.118.4 Member Function Documentation	812
6.118.4.1 is_const()	812
6.118.4.2 is_void()	812
6.118.4.3 nargs()	813
6.118.4.4 operator>()	813
6.118.4.5 signature()	814
6.118.5 Member Data Documentation	814
6.118.5.1 parent_method_pointer	814
6.119 Rcpp::CppInheritedProperty< Class, Parent > Class Template Reference	815
6.119.1 Detailed Description	816
6.119.2 Member Typedef Documentation	816
6.119.2.1 Base	816
6.119.3 Constructor & Destructor Documentation	816
6.119.3.1 CppInheritedProperty()	816
6.119.4 Member Function Documentation	817
6.119.4.1 get()	817
6.119.4.2 get_class()	817
6.119.4.3 is_readonly()	818
6.119.4.4 set()	818
6.119.5 Member Data Documentation	819
6.119.5.1 parent_property	819
6.120 Rcpp::CppMethod< Class > Class Template Reference	819
6.120.1 Detailed Description	820
6.120.2 Member Typedef Documentation	820
6.120.2.1 XP	820
6.120.3 Constructor & Destructor Documentation	820
6.120.3.1 CppMethod()	820
6.120.3.2 ~CppMethod()	821
6.120.4 Member Function Documentation	821
6.120.4.1 is_const()	821
6.120.4.2 is_void()	821
6.120.4.3 nargs()	821
6.120.4.4 operator>()	822
6.120.4.5 signature()	822
6.121 Rcpp::CppObject Class Reference	822

---

6.121.1 Detailed Description	823
6.121.2 Member Typedef Documentation	823
6.121.2.1 Base	823
6.121.2.2 XP	824
6.121.3 Constructor & Destructor Documentation	824
6.121.3.1 CppObject()	824
6.122 Rcpp::attributes::CppPackageIncludeGenerator Class Reference	824
6.122.1 Detailed Description	825
6.122.2 Constructor & Destructor Documentation	825
6.122.2.1 CppPackageIncludeGenerator()	826
6.122.3 Member Function Documentation	826
6.122.3.1 commit()	826
6.122.3.2 doWriteFunctions()	827
6.122.3.3 getHeaderGuard()	827
6.122.3.4 writeBegin()	827
6.122.3.5 writeEnd()	828
6.122.4 Member Data Documentation	828
6.122.4.1 includeDir_	828
6.123 Rcpp::CppProperty< Class > Class Template Reference	829
6.123.1 Detailed Description	829
6.123.2 Member Typedef Documentation	830
6.123.2.1 XP	830
6.123.3 Constructor & Destructor Documentation	830
6.123.3.1 CppProperty()	830
6.123.3.2 ~CppProperty()	830
6.123.4 Member Function Documentation	830
6.123.4.1 get()	830
6.123.4.2 get_class()	831
6.123.4.3 is_readonly()	831
6.123.4.4 set()	831
6.123.5 Member Data Documentation	831
6.123.5.1 docstring	832
6.124 CppProperty_GetConstMethod< Class, PROP > Class Template Reference	832
6.124.1 Detailed Description	833
6.124.2 Member Typedef Documentation	833
6.124.2.1 GetMethod	833
6.124.2.2 prop_class	833
6.124.3 Constructor & Destructor Documentation	834
6.124.3.1 CppProperty_GetConstMethod()	834

---

6.124.4 Member Function Documentation	834
6.124.4.1 get()	834
6.124.4.2 get_class()	835
6.124.4.3 is_readonly()	835
6.124.4.4 set()	835
6.124.5 Member Data Documentation	835
6.124.5.1 class_name	835
6.124.5.2 getter	836
6.125 CppProperty_GetConstMethod_SetMethod< Class, PROP > Class Template Reference	836
6.125.1 Detailed Description	837
6.125.2 Member Typedef Documentation	837
6.125.2.1 GetMethod	837
6.125.2.2 prop_class	838
6.125.2.3 SetMethod	838
6.125.3 Constructor & Destructor Documentation	838
6.125.3.1 CppProperty_GetConstMethod_SetMethod()	838
6.125.4 Member Function Documentation	838
6.125.4.1 get()	839
6.125.4.2 get_class()	839
6.125.4.3 is_readonly()	839
6.125.4.4 set()	840
6.125.5 Member Data Documentation	840
6.125.5.1 class_name	840
6.125.5.2 getter	840
6.125.5.3 setter	841
6.126 CppProperty_GetConstMethod_SetPointer< Class, PROP > Class Template Reference	841
6.126.1 Detailed Description	842
6.126.2 Member Typedef Documentation	842
6.126.2.1 GetMethod	842
6.126.2.2 prop_class	843
6.126.2.3 SetMethod	843
6.126.3 Constructor & Destructor Documentation	843
6.126.3.1 CppProperty_GetConstMethod_SetPointer()	843
6.126.4 Member Function Documentation	843
6.126.4.1 get()	844
6.126.4.2 get_class()	844
6.126.4.3 is_readonly()	844
6.126.4.4 set()	845
6.126.5 Member Data Documentation	845

---

6.126.5.1 class_name	845
6.126.5.2 getter	845
6.126.5.3 setter	846
6.127 CppProperty_GetMethod< Class, PROP > Class Template Reference	846
6.127.1 Detailed Description	847
6.127.2 Member Typedef Documentation	847
6.127.2.1 GetMethod	847
6.127.2.2 prop_class	847
6.127.3 Constructor & Destructor Documentation	848
6.127.3.1 CppProperty_GetMethod()	848
6.127.4 Member Function Documentation	848
6.127.4.1 get()	848
6.127.4.2 get_class()	849
6.127.4.3 is_readonly()	849
6.127.4.4 set()	849
6.127.5 Member Data Documentation	849
6.127.5.1 class_name	849
6.127.5.2 getter	850
6.128 CppProperty_GetMethod_SetMethod< Class, PROP > Class Template Reference	850
6.128.1 Detailed Description	851
6.128.2 Member Typedef Documentation	851
6.128.2.1 GetMethod	851
6.128.2.2 prop_class	852
6.128.2.3 SetMethod	852
6.128.3 Constructor & Destructor Documentation	852
6.128.3.1 CppProperty_GetMethod_SetMethod()	852
6.128.4 Member Function Documentation	852
6.128.4.1 get()	853
6.128.4.2 get_class()	853
6.128.4.3 is_readonly()	853
6.128.4.4 set()	854
6.128.5 Member Data Documentation	854
6.128.5.1 class_name	854
6.128.5.2 getter	854
6.128.5.3 setter	855
6.129 CppProperty_GetMethod_SetPointer< Class, PROP > Class Template Reference	855
6.129.1 Detailed Description	856
6.129.2 Member Typedef Documentation	856
6.129.2.1 GetMethod	856

6.129.2.2 prop_class	857
6.129.2.3 SetMethod	857
6.129.3 Constructor & Destructor Documentation	857
6.129.3.1 CppProperty_GetMethod_SetPointer()	857
6.129.4 Member Function Documentation	857
6.129.4.1 get()	858
6.129.4.2 get_class()	858
6.129.4.3 is_readonly()	858
6.129.4.4 set()	859
6.129.5 Member Data Documentation	859
6.129.5.1 class_name	859
6.129.5.2 getter	859
6.129.5.3 setter	860
6.130 CppProperty_GetPointer_SetMethod< Class, PROP > Class Template Reference	860
6.130.1 Detailed Description	861
6.130.2 Member Typedef Documentation	861
6.130.2.1 GetMethod	861
6.130.2.2 prop_class	862
6.130.2.3 SetMethod	862
6.130.3 Constructor & Destructor Documentation	862
6.130.3.1 CppProperty_GetPointer_SetMethod()	862
6.130.4 Member Function Documentation	862
6.130.4.1 get()	863
6.130.4.2 get_class()	863
6.130.4.3 is_readonly()	863
6.130.4.4 set()	864
6.130.5 Member Data Documentation	864
6.130.5.1 class_name	864
6.130.5.2 getter	864
6.130.5.3 setter	865
6.131 CppProperty_GetPointer_SetPointer< Class, PROP > Class Template Reference	865
6.131.1 Detailed Description	866
6.131.2 Member Typedef Documentation	866
6.131.2.1 GetMethod	866
6.131.2.2 prop_class	867
6.131.2.3 SetMethod	867
6.131.3 Constructor & Destructor Documentation	867
6.131.3.1 CppProperty_GetPointer_SetPointer()	867
6.131.4 Member Function Documentation	867

---

6.131.4.1	get()	868
6.131.4.2	get_class()	868
6.131.4.3	is_readonly()	868
6.131.4.4	set()	869
6.131.5	Member Data Documentation	869
6.131.5.1	class_name	869
6.131.5.2	getter	869
6.131.5.3	setter	870
6.132	CppProperty_GetPointerMethod< Class, PROP > Class Template Reference	870
6.132.1	Detailed Description	871
6.132.2	Member Typedef Documentation	871
6.132.2.1	GetMethod	871
6.132.2.2	prop_class	871
6.132.3	Constructor & Destructor Documentation	872
6.132.3.1	CppProperty_GetPointerMethod()	872
6.132.4	Member Function Documentation	872
6.132.4.1	get()	872
6.132.4.2	get_class()	873
6.132.4.3	is_readonly()	873
6.132.4.4	set()	873
6.132.5	Member Data Documentation	873
6.132.5.1	class_name	873
6.132.5.2	getter	874
6.133	CppProperty_Getter< PROP > Class Template Reference	874
6.133.1	Detailed Description	875
6.133.2	Member Typedef Documentation	875
6.133.2.1	pointer	875
6.133.2.2	prop_class	875
6.133.3	Constructor & Destructor Documentation	876
6.133.3.1	CppProperty_Getter()	876
6.133.4	Member Function Documentation	876
6.133.4.1	get()	876
6.133.4.2	get_class()	877
6.133.4.3	is_readonly()	877
6.133.4.4	set()	877
6.133.5	Member Data Documentation	877
6.133.5.1	class_name	877
6.133.5.2	ptr	878
6.134	CppProperty_Getter_Setter< PROP > Class Template Reference	878

---

6.134.1 Detailed Description	879
6.134.2 Member Typedef Documentation	879
6.134.2.1 pointer	879
6.134.2.2 prop_class	879
6.134.3 Constructor & Destructor Documentation	880
6.134.3.1 CppProperty_Getter_Setter()	880
6.134.4 Member Function Documentation	880
6.134.4.1 get()	880
6.134.4.2 get_class()	881
6.134.4.3 is_readonly()	881
6.134.4.4 set()	881
6.134.5 Member Data Documentation	881
6.134.5.1 class_name	881
6.134.5.2 ptr	882
6.135 Rcpp::algorithm::helpers::ctype< T > Struct Template Reference	882
6.135.1 Detailed Description	882
6.135.2 Member Typedef Documentation	883
6.135.2.1 type	883
6.135.3 Member Function Documentation	883
6.135.3.1 make()	883
6.135.3.2 test() [1/13]	883
6.135.3.3 test() [2/13]	883
6.135.3.4 test() [3/13]	883
6.135.3.5 test() [4/13]	884
6.135.3.6 test() [5/13]	884
6.135.3.7 test() [6/13]	884
6.135.3.8 test() [7/13]	884
6.135.3.9 test() [8/13]	884
6.135.3.10 test() [9/13]	884
6.135.3.11 test() [10/13]	885
6.135.3.12 test() [11/13]	885
6.135.3.13 test() [12/13]	885
6.135.3.14 test() [13/13]	885
6.136 Rcpp::algorithm::helpers::CTYPE_CHAR Struct Reference	885
6.136.1 Detailed Description	886
6.136.2 Member Data Documentation	886
6.136.2.1 a	886
6.137 Rcpp::algorithm::helpers::CTYPE_DOUBLE Struct Reference	886
6.137.1 Detailed Description	886

6.137.2 Member Data Documentation	886
6.137.2.1 a	886
6.138 Rcpp::algorithm::helpers::CTYPE_FLOAT Struct Reference	887
6.138.1 Detailed Description	887
6.138.2 Member Data Documentation	887
6.138.2.1 a	887
6.139 Rcpp::algorithm::helpers::ctype_helper< I > Struct Template Reference	887
6.139.1 Detailed Description	887
6.139.2 Member Data Documentation	888
6.139.2.1 value	888
6.140 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_CHAR) > Struct Reference	888
6.140.1 Detailed Description	888
6.140.2 Member Typedef Documentation	888
6.140.2.1 type	888
6.140.3 Member Data Documentation	889
6.140.3.1 value	889
6.141 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_DOUBLE) > Struct Reference	889
6.141.1 Detailed Description	889
6.141.2 Member Typedef Documentation	889
6.141.2.1 type	889
6.141.3 Member Data Documentation	890
6.141.3.1 value	890
6.142 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_FLOAT) > Struct Reference	890
6.142.1 Detailed Description	890
6.142.2 Member Typedef Documentation	890
6.142.2.1 type	890
6.142.3 Member Data Documentation	891
6.142.3.1 value	891
6.143 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_INT) > Struct Reference	891
6.143.1 Detailed Description	891
6.143.2 Member Typedef Documentation	891
6.143.2.1 type	891
6.143.3 Member Data Documentation	892
6.143.3.1 value	892
6.144 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG) > Struct Reference	892
6.144.1 Detailed Description	892
6.144.2 Member Typedef Documentation	892
6.144.2.1 type	892
6.144.3 Member Data Documentation	893



---

6.144.3.1 value	893
6.145 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG_DOUBLE) > Struct Reference	893
6.145.1 Detailed Description	893
6.145.2 Member Typedef Documentation	893
6.145.2.1 type	893
6.145.3 Member Data Documentation	894
6.145.3.1 value	894
6.146 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_SHORT) > Struct Reference	894
6.146.1 Detailed Description	894
6.146.2 Member Typedef Documentation	894
6.146.2.1 type	894
6.146.3 Member Data Documentation	895
6.146.3.1 value	895
6.147 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_STRING) > Struct Reference	895
6.147.1 Detailed Description	895
6.147.2 Member Typedef Documentation	895
6.147.2.1 type	895
6.147.3 Member Data Documentation	896
6.147.3.1 value	896
6.148 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) > Struct Reference	896
6.148.1 Detailed Description	896
6.148.2 Member Typedef Documentation	896
6.148.2.1 type	896
6.148.3 Member Data Documentation	897
6.148.3.1 value	897
6.149 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_INT) > Struct Reference	897
6.149.1 Detailed Description	897
6.149.2 Member Typedef Documentation	897
6.149.2.1 type	897
6.149.3 Member Data Documentation	898
6.149.3.1 value	898
6.150 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) > Struct Reference	898
6.150.1 Detailed Description	898
6.150.2 Member Typedef Documentation	898
6.150.2.1 type	898
6.150.3 Member Data Documentation	899
6.150.3.1 value	899
6.151 Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) > Struct Reference	899
6.151.1 Detailed Description	899

---

6.151.2 Member Typedef Documentation	899
6.151.2.1 type	899
6.151.3 Member Data Documentation	900
6.151.3.1 value	900
6.152 Rcpp::algorithm::helpers::CTYPE_INT Struct Reference	900
6.152.1 Detailed Description	900
6.152.2 Member Data Documentation	900
6.152.2.1 a	900
6.153 Rcpp::algorithm::helpers::CTYPE_LONG Struct Reference	901
6.153.1 Detailed Description	901
6.153.2 Member Data Documentation	901
6.153.2.1 a	901
6.154 Rcpp::algorithm::helpers::CTYPE_LONG_DOUBLE Struct Reference	901
6.154.1 Detailed Description	901
6.154.2 Member Data Documentation	902
6.154.2.1 a	902
6.155 Rcpp::algorithm::helpers::CTYPE_SHORT Struct Reference	902
6.155.1 Detailed Description	902
6.155.2 Member Data Documentation	902
6.155.2.1 a	902
6.156 Rcpp::algorithm::helpers::CTYPE_STRING Struct Reference	903
6.156.1 Detailed Description	903
6.156.2 Member Data Documentation	903
6.156.2.1 a	903
6.157 Rcpp::algorithm::helpers::CTYPE_UNKNOWN Struct Reference	903
6.157.1 Detailed Description	903
6.157.2 Member Data Documentation	904
6.157.2.1 a	904
6.158 Rcpp::algorithm::helpers::CTYPE_UNSIGNED_CHAR Struct Reference	904
6.158.1 Detailed Description	904
6.158.2 Member Data Documentation	904
6.158.2.1 a	904
6.159 Rcpp::algorithm::helpers::CTYPE_UNSIGNED_INT Struct Reference	905
6.159.1 Detailed Description	905
6.159.2 Member Data Documentation	905
6.159.2.1 a	905
6.160 Rcpp::algorithm::helpers::CTYPE_UNSIGNED_LONG Struct Reference	905
6.160.1 Detailed Description	905
6.160.2 Member Data Documentation	906

---

6.160.2.1 a	906
6.161 Rcpp::algorithm::helpers::CTYPE_UNSIGNED_SHORT Struct Reference	906
6.161.1 Detailed Description	906
6.161.2 Member Data Documentation	906
6.161.2.1 a	906
6.162 Rcpp::sugar::Cummax< RTYPE, NA, T > Class Template Reference	907
6.162.1 Detailed Description	908
6.162.2 Member Typedef Documentation	908
6.162.2.1 STORAGE	908
6.162.2.2 VEC_TYPE	908
6.162.2.3 VECTOR	908
6.162.3 Constructor & Destructor Documentation	908
6.162.3.1 Cummax()	909
6.162.4 Member Function Documentation	909
6.162.4.1 get()	909
6.162.5 Member Data Documentation	909
6.162.5.1 object	909
6.163 Rcpp::sugar::Cummin< RTYPE, NA, T > Class Template Reference	910
6.163.1 Detailed Description	911
6.163.2 Member Typedef Documentation	911
6.163.2.1 STORAGE	911
6.163.2.2 VEC_TYPE	911
6.163.2.3 VECTOR	911
6.163.3 Constructor & Destructor Documentation	911
6.163.3.1 Cummin()	912
6.163.4 Member Function Documentation	912
6.163.4.1 get()	912
6.163.5 Member Data Documentation	912
6.163.5.1 object	912
6.164 Rcpp::sugar::Cumprod< RTYPE, NA, T > Class Template Reference	913
6.164.1 Detailed Description	914
6.164.2 Member Typedef Documentation	914
6.164.2.1 STORAGE	914
6.164.2.2 VEC_TYPE	914
6.164.2.3 VECTOR	914
6.164.3 Constructor & Destructor Documentation	914
6.164.3.1 Cumprod()	915
6.164.4 Member Function Documentation	915
6.164.4.1 get()	915

---

6.164.5 Member Data Documentation	915
6.164.5.1 object	915
6.165 Rcpp::sugar::Cumsum< RTYPE, NA, T > Class Template Reference	916
6.165.1 Detailed Description	917
6.165.2 Member Typedef Documentation	917
6.165.2.1 STORAGE	917
6.165.2.2 VEC_TYPE	917
6.165.2.3 VECTOR	917
6.165.3 Constructor & Destructor Documentation	917
6.165.3.1 Cumsum()	918
6.165.4 Member Function Documentation	918
6.165.4.1 get()	918
6.165.5 Member Data Documentation	918
6.165.5.1 object	918
6.166 Rcpp::stats::D0< RTYPE, NA, T > Class Template Reference	919
6.166.1 Detailed Description	920
6.166.2 Member Typedef Documentation	920
6.166.2.1 FunPtr	920
6.166.2.2 VEC_TYPE	920
6.166.3 Constructor & Destructor Documentation	920
6.166.3.1 D0()	920
6.166.4 Member Function Documentation	921
6.166.4.1 operator[]()	921
6.166.4.2 size()	921
6.166.5 Member Data Documentation	921
6.166.5.1 log	922
6.166.5.2 ptr	922
6.166.5.3 vec	922
6.167 Rcpp::stats::D1< RTYPE, NA, T > Class Template Reference	923
6.167.1 Detailed Description	924
6.167.2 Member Typedef Documentation	924
6.167.2.1 FunPtr	924
6.167.2.2 VEC_TYPE	924
6.167.3 Constructor & Destructor Documentation	924
6.167.3.1 D1()	925
6.167.4 Member Function Documentation	925
6.167.4.1 operator[]()	925
6.167.4.2 size()	925
6.167.5 Member Data Documentation	926

---

6.167.5.1 log	926
6.167.5.2 p0	926
6.167.5.3 ptr	926
6.167.5.4 vec	926
6.168 Rcpp::stats::D2< RTYPE, NA, T > Class Template Reference	927
6.168.1 Detailed Description	928
6.168.2 Member Typedef Documentation	928
6.168.2.1 FunPtr	928
6.168.2.2 VEC_TYPE	928
6.168.3 Constructor & Destructor Documentation	928
6.168.3.1 D2()	929
6.168.4 Member Function Documentation	929
6.168.4.1 operator[]()	929
6.168.4.2 size()	929
6.168.5 Member Data Documentation	930
6.168.5.1 log	930
6.168.5.2 p0	930
6.168.5.3 p1	930
6.168.5.4 ptr	930
6.168.5.5 vec	931
6.169 Rcpp::stats::D3< RTYPE, NA, T > Class Template Reference	931
6.169.1 Detailed Description	932
6.169.2 Member Typedef Documentation	932
6.169.2.1 FunPtr	932
6.169.2.2 VEC_TYPE	933
6.169.3 Constructor & Destructor Documentation	933
6.169.3.1 D3()	933
6.169.4 Member Function Documentation	933
6.169.4.1 operator[]()	933
6.169.4.2 size()	934
6.169.5 Member Data Documentation	934
6.169.5.1 log	934
6.169.5.2 p0	934
6.169.5.3 p1	935
6.169.5.4 p2	935
6.169.5.5 ptr	935
6.169.5.6 vec	935
6.170 Rcpp::DataFrame_Impl< StoragePolicy > Class Template Reference	936
6.170.1 Detailed Description	937

---

6.170.2 Member Typedef Documentation	937
6.170.2.1 Parent	938
6.170.3 Constructor & Destructor Documentation	938
6.170.3.1 DataFrame_Impl() [1/4]	938
6.170.3.2 DataFrame_Impl() [2/4]	938
6.170.3.3 DataFrame_Impl() [3/4]	939
6.170.3.4 DataFrame_Impl() [4/4]	939
6.170.4 Member Function Documentation	939
6.170.4.1 cols()	940
6.170.4.2 create()	940
6.170.4.3 from_list()	941
6.170.4.4 ncol()	941
6.170.4.5 nrow()	942
6.170.4.6 nrows()	942
6.170.4.7 operator=() [1/2]	942
6.170.4.8 operator=() [2/2]	943
6.170.4.9 push_back() [1/2]	943
6.170.4.10 push_back() [2/2]	944
6.170.4.11 push_front() [1/2]	944
6.170.4.12 push_front() [2/2]	945
6.170.4.13 rows()	945
6.170.4.14 set__()	946
6.170.4.15 set_type_after_push()	946
6.171 Rcpp::Date Class Reference	947
6.171.1 Detailed Description	948
6.171.2 Constructor & Destructor Documentation	948
6.171.2.1 Date() [1/6]	948
6.171.2.2 Date() [2/6]	948
6.171.2.3 Date() [3/6]	949
6.171.2.4 Date() [4/6]	949
6.171.2.5 Date() [5/6]	950
6.171.2.6 Date() [6/6]	950
6.171.3 Member Function Documentation	951
6.171.3.1 baseYear()	951
6.171.3.2 format()	951
6.171.3.3 getDate()	951
6.171.3.4 getDay()	952
6.171.3.5 getMonth()	952
6.171.3.6 getWeekday()	952

---

6.171.3.7	getYear()	952
6.171.3.8	getYearday()	952
6.171.3.9	is_na()	953
6.171.3.10	operator double()	953
6.171.3.11	update_tm()	953
6.171.4	Friends And Related Function Documentation	954
6.171.4.1	operator"!="	954
6.171.4.2	operator+	954
6.171.4.3	operator-	954
6.171.4.4	operator<	954
6.171.4.5	operator<<	955
6.171.4.6	operator<=	955
6.171.4.7	operator==	955
6.171.4.8	operator>	955
6.171.4.9	operator>=	955
6.171.5	Member Data Documentation	956
6.171.5.1	m_d	956
6.171.5.2	m_tm	956
6.172	Rcpp::Datetime Class Reference	956
6.172.1	Detailed Description	957
6.172.2	Constructor & Destructor Documentation	957
6.172.2.1	Datetime() [1/4]	958
6.172.2.2	Datetime() [2/4]	958
6.172.2.3	Datetime() [3/4]	959
6.172.2.4	Datetime() [4/4]	959
6.172.3	Member Function Documentation	959
6.172.3.1	baseYear()	960
6.172.3.2	format()	960
6.172.3.3	getDay()	960
6.172.3.4	getFractionalTimestamp()	960
6.172.3.5	getHours()	960
6.172.3.6	getMicroSeconds()	961
6.172.3.7	getMinutes()	961
6.172.3.8	getMonth()	961
6.172.3.9	getSeconds()	961
6.172.3.10	getWeekday()	961
6.172.3.11	getYear()	962
6.172.3.12	getYearday()	962
6.172.3.13	is_na()	962

---

6.172.3.14 operator double()	962
6.172.3.15 update_tm()	963
6.172.4 Friends And Related Function Documentation	963
6.172.4.1 operator"!="	963
6.172.4.2 operator+ [1/2]	963
6.172.4.3 operator+ [2/2]	964
6.172.4.4 operator-	964
6.172.4.5 operator<	964
6.172.4.6 operator<<	964
6.172.4.7 operator<=	964
6.172.4.8 operator==	965
6.172.4.9 operator>	965
6.172.4.10 operator>=	965
6.172.5 Member Data Documentation	965
6.172.5.1 m_dt	965
6.172.5.2 m_tm	966
6.172.5.3 m_us	966
6.173 Rcpp::algorithm::helpers::decays_to_ctype< T > Struct Template Reference	966
6.173.1 Detailed Description	967
6.173.2 Member Function Documentation	967
6.173.2.1 make()	967
6.173.2.2 test() [1/13]	967
6.173.2.3 test() [2/13]	967
6.173.2.4 test() [3/13]	967
6.173.2.5 test() [4/13]	968
6.173.2.6 test() [5/13]	968
6.173.2.7 test() [6/13]	968
6.173.2.8 test() [7/13]	968
6.173.2.9 test() [8/13]	968
6.173.2.10 test() [9/13]	968
6.173.2.11 test() [10/13]	969
6.173.2.12 test() [11/13]	969
6.173.2.13 test() [12/13]	969
6.173.2.14 test() [13/13]	969
6.173.3 Member Data Documentation	969
6.173.3.1 value	969
6.174 Rcpp::sugar::Diag_Extractor< RTYPE, NA, T > Class Template Reference	970
6.174.1 Detailed Description	971
6.174.2 Member Typedef Documentation	971



---

6.174.2.1 MAT_TYPE	971
6.174.2.2 STORAGE	971
6.174.3 Constructor & Destructor Documentation	971
6.174.3.1 Diag_Extractor()	972
6.174.4 Member Function Documentation	972
6.174.4.1 operator[]()	972
6.174.4.2 size()	972
6.174.5 Member Data Documentation	972
6.174.5.1 n	972
6.174.5.2 object	973
6.175 Rcpp::sugar::Diag_Maker< RTYPE, NA, T > Class Template Reference	973
6.175.1 Detailed Description	974
6.175.2 Member Typedef Documentation	974
6.175.2.1 STORAGE	974
6.175.2.2 VEC_TYPE	975
6.175.3 Constructor & Destructor Documentation	975
6.175.3.1 Diag_Maker()	975
6.175.4 Member Function Documentation	975
6.175.4.1 ncol()	975
6.175.4.2 nrow()	975
6.175.4.3 operator>()()	976
6.175.4.4 size()	976
6.175.5 Member Data Documentation	976
6.175.5.1 n	976
6.175.5.2 object	976
6.176 Rcpp::sugar::diag_result_type_trait< T > Struct Template Reference	977
6.176.1 Detailed Description	977
6.176.2 Member Typedef Documentation	977
6.176.2.1 type	977
6.177 Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T > Class Template Reference	978
6.177.1 Detailed Description	979
6.177.2 Member Typedef Documentation	979
6.177.2.1 LHS_TYPE	979
6.177.2.2 STORAGE	979
6.177.3 Constructor & Destructor Documentation	980
6.177.3.1 Diff()	980
6.177.4 Member Function Documentation	980
6.177.4.1 operator[]()	980
6.177.4.2 set_previous()	981

---

6.177.4.3 size()	981
6.177.5 Member Data Documentation	981
6.177.5.1 lhs	982
6.177.5.2 previous	982
6.177.5.3 previous_index	982
6.177.5.4 was_na	982
6.178 Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T > Class Template Reference	983
6.178.1 Detailed Description	984
6.178.2 Member Typedef Documentation	984
6.178.2.1 LHS_TYPE	984
6.178.3 Constructor & Destructor Documentation	984
6.178.3.1 Diff()	984
6.178.4 Member Function Documentation	984
6.178.4.1 operator[]()	985
6.178.4.2 size()	985
6.178.5 Member Data Documentation	985
6.178.5.1 lhs	985
6.178.5.2 previous	986
6.178.5.3 previous_index	986
6.179 Rcpp::sugar::Diff< RTYPE, false, LHS_T > Class Template Reference	986
6.179.1 Detailed Description	987
6.179.2 Member Typedef Documentation	987
6.179.2.1 LHS_TYPE	988
6.179.2.2 STORAGE	988
6.179.3 Constructor & Destructor Documentation	988
6.179.3.1 Diff()	988
6.179.4 Member Function Documentation	988
6.179.4.1 operator[]()	989
6.179.4.2 size()	989
6.179.5 Member Data Documentation	990
6.179.5.1 lhs	990
6.179.5.2 previous	990
6.179.5.3 previous_index	990
6.180 Rcpp::Dimension Class Reference	990
6.180.1 Detailed Description	991
6.180.2 Member Typedef Documentation	991
6.180.2.1 const_reference	991
6.180.2.2 reference	991
6.180.3 Constructor & Destructor Documentation	991

---

6.180.3.1 Dimension() [1/6]	992
6.180.3.2 Dimension() [2/6]	992
6.180.3.3 Dimension() [3/6]	992
6.180.3.4 Dimension() [4/6]	992
6.180.3.5 Dimension() [5/6]	992
6.180.3.6 Dimension() [6/6]	993
6.180.4 Member Function Documentation	993
6.180.4.1 operator SEXP()	993
6.180.4.2 operator=()	993
6.180.4.3 operator[]() [1/2]	994
6.180.4.4 operator[]() [2/2]	994
6.180.4.5 prod()	994
6.180.4.6 size()	994
6.180.5 Member Data Documentation	994
6.180.5.1 dims	995
6.181 Rcpp::internal::DimNameProxy Class Reference	995
6.181.1 Detailed Description	995
6.181.2 Constructor & Destructor Documentation	995
6.181.2.1 DimNameProxy() [1/2]	996
6.181.2.2 DimNameProxy() [2/2]	996
6.181.3 Member Function Documentation	996
6.181.3.1 assign()	996
6.181.3.2 operator SEXP()	997
6.181.3.3 operator T()	997
6.181.3.4 operator=() [1/2]	997
6.181.3.5 operator=() [2/2]	998
6.181.4 Member Data Documentation	998
6.181.4.1 data_	998
6.181.4.2 dim_	998
6.182 Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T > Class Template Reference	999
6.182.1 Detailed Description	1000
6.182.2 Member Typedef Documentation	1001
6.182.2.1 STORAGE	1001
6.182.2.2 VEC_EXT	1001
6.182.2.3 VEC_TYPE	1001
6.182.3 Constructor & Destructor Documentation	1001
6.182.3.1 Divides_Primitive_Vector()	1001
6.182.4 Member Function Documentation	1002
6.182.4.1 operator[]()	1002

---

6.182.4.2 size()	1002
6.182.5 Member Data Documentation	1002
6.182.5.1 lhs	1002
6.182.5.2 lhs_na	1003
6.182.5.3 rhs	1003
6.183 Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T > Class Template Reference	1003
6.183.1 Detailed Description	1005
6.183.2 Member Typedef Documentation	1005
6.183.2.1 VEC_EXT	1005
6.183.2.2 VEC_TYPE	1005
6.183.3 Constructor & Destructor Documentation	1006
6.183.3.1 Divides_Primitive_Vector()	1006
6.183.4 Member Function Documentation	1006
6.183.4.1 operator[]()	1006
6.183.4.2 size()	1006
6.183.5 Member Data Documentation	1006
6.183.5.1 lhs	1007
6.183.5.2 rhs	1007
6.184 Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T > Class Template Reference	1007
6.184.1 Detailed Description	1008
6.184.2 Member Typedef Documentation	1009
6.184.2.1 VEC_EXT	1009
6.184.2.2 VEC_TYPE	1009
6.184.3 Constructor & Destructor Documentation	1009
6.184.3.1 Divides_Primitive_Vector()	1009
6.184.4 Member Function Documentation	1009
6.184.4.1 operator[]()	1010
6.184.4.2 size()	1010
6.184.5 Member Data Documentation	1010
6.184.5.1 lhs	1010
6.184.5.2 rhs	1010
6.185 Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T > Class Template Reference	1011
6.185.1 Detailed Description	1012
6.185.2 Member Typedef Documentation	1013
6.185.2.1 STORAGE	1013
6.185.2.2 VEC_EXT	1013
6.185.2.3 VEC_TYPE	1013
6.185.3 Constructor & Destructor Documentation	1013
6.185.3.1 Divides_Primitive_Vector()	1013

---

6.185.4 Member Function Documentation	1014
6.185.4.1 operator[]()	1014
6.185.4.2 size()	1014
6.185.5 Member Data Documentation	1014
6.185.5.1 lhs	1014
6.185.5.2 lhs_na	1014
6.185.5.3 rhs	1015
6.186 Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T > Class Template Reference	1015
6.186.1 Detailed Description	1016
6.186.2 Member Typedef Documentation	1017
6.186.2.1 STORAGE	1017
6.186.2.2 VEC_EXT	1017
6.186.2.3 VEC_TYPE	1017
6.186.3 Constructor & Destructor Documentation	1017
6.186.3.1 Divides_Vector_Primitive()	1017
6.186.4 Member Function Documentation	1018
6.186.4.1 operator[]()	1018
6.186.4.2 size()	1018
6.186.5 Member Data Documentation	1018
6.186.5.1 lhs	1018
6.186.5.2 rhs	1019
6.186.5.3 rhs_na	1019
6.187 Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T > Class Template Reference	1019
6.187.1 Detailed Description	1021
6.187.2 Member Typedef Documentation	1021
6.187.2.1 VEC_EXT	1021
6.187.2.2 VEC_TYPE	1021
6.187.3 Constructor & Destructor Documentation	1022
6.187.3.1 Divides_Vector_Primitive()	1022
6.187.4 Member Function Documentation	1022
6.187.4.1 operator[]()	1022
6.187.4.2 size()	1022
6.187.5 Member Data Documentation	1022
6.187.5.1 lhs	1023
6.187.5.2 rhs	1023
6.188 Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T > Class Template Reference	1023
6.188.1 Detailed Description	1024
6.188.2 Member Typedef Documentation	1025
6.188.2.1 VEC_EXT	1025

---

6.188.2.2 VEC_TYPE . . . . .	1025
6.188.3 Constructor & Destructor Documentation . . . . .	1025
6.188.3.1 Divides_Vector_Primitive() . . . . .	1025
6.188.4 Member Function Documentation . . . . .	1025
6.188.4.1 operator[]() . . . . .	1026
6.188.4.2 size() . . . . .	1026
6.188.5 Member Data Documentation . . . . .	1026
6.188.5.1 lhs . . . . .	1026
6.188.5.2 rhs . . . . .	1026
6.189 Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T > Class Template Reference . . . . .	1027
6.189.1 Detailed Description . . . . .	1028
6.189.2 Member Typedef Documentation . . . . .	1029
6.189.2.1 STORAGE . . . . .	1029
6.189.2.2 VEC_EXT . . . . .	1029
6.189.2.3 VEC_TYPE . . . . .	1029
6.189.3 Constructor & Destructor Documentation . . . . .	1029
6.189.3.1 Divides_Vector_Primitive() . . . . .	1029
6.189.4 Member Function Documentation . . . . .	1030
6.189.4.1 operator[]() . . . . .	1030
6.189.4.2 size() . . . . .	1030
6.189.5 Member Data Documentation . . . . .	1030
6.189.5.1 lhs . . . . .	1030
6.189.5.2 rhs . . . . .	1030
6.189.5.3 rhs_na . . . . .	1031
6.190 Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference . . . . .	1031
6.190.1 Detailed Description . . . . .	1032
6.190.2 Member Typedef Documentation . . . . .	1032
6.190.2.1 LHS_EXT . . . . .	1032
6.190.2.2 LHS_TYPE . . . . .	1032
6.190.2.3 RHS_EXT . . . . .	1033
6.190.2.4 RHS_TYPE . . . . .	1033
6.190.2.5 STORAGE . . . . .	1033
6.190.3 Constructor & Destructor Documentation . . . . .	1033
6.190.3.1 Divides_Vector_Vector() . . . . .	1033
6.190.4 Member Function Documentation . . . . .	1033
6.190.4.1 operator[]() . . . . .	1034
6.190.4.2 size() . . . . .	1034
6.190.5 Member Data Documentation . . . . .	1034

---

6.190.5.1 lhs	1034
6.190.5.2 rhs	1035
6.191 Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > Class Template Reference	1035
6.191.1 Detailed Description	1036
6.191.2 Member Typedef Documentation	1036
6.191.2.1 LHS_EXT	1036
6.191.2.2 LHS_TYPE	1037
6.191.2.3 RHS_EXT	1037
6.191.2.4 RHS_TYPE	1037
6.191.3 Constructor & Destructor Documentation	1037
6.191.3.1 Divides_Vector_Vector()	1037
6.191.4 Member Function Documentation	1037
6.191.4.1 operator[]()	1038
6.191.4.2 size()	1038
6.191.5 Member Data Documentation	1038
6.191.5.1 lhs	1038
6.191.5.2 rhs	1038
6.192 Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > Class Template Reference	1039
6.192.1 Detailed Description	1040
6.192.2 Member Typedef Documentation	1040
6.192.2.1 LHS_EXT	1040
6.192.2.2 LHS_TYPE	1040
6.192.2.3 RHS_EXT	1040
6.192.2.4 RHS_TYPE	1041
6.192.3 Constructor & Destructor Documentation	1041
6.192.3.1 Divides_Vector_Vector()	1041
6.192.4 Member Function Documentation	1041
6.192.4.1 operator[]()	1041
6.192.4.2 size()	1041
6.192.5 Member Data Documentation	1042
6.192.5.1 lhs	1042
6.192.5.2 rhs	1042
6.193 Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > Class Template Reference	1042
6.193.1 Detailed Description	1043
6.193.2 Member Typedef Documentation	1043
6.193.2.1 LHS_EXT	1043
6.193.2.2 LHS_TYPE	1044

---

6.193.2.3 RHS_EXT	1044
6.193.2.4 RHS_TYPE	1044
6.193.3 Constructor & Destructor Documentation	1044
6.193.3.1 Divides_Vector_Vector()	1044
6.193.4 Member Function Documentation	1044
6.193.4.1 operator[]()	1045
6.193.4.2 size()	1045
6.193.5 Member Data Documentation	1045
6.193.5.1 lhs	1045
6.193.5.2 rhs	1045
6.194 Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	1046
6.194.1 Detailed Description	1047
6.194.2 Member Typedef Documentation	1047
6.194.2.1 LHS_EXT	1047
6.194.2.2 LHS_TYPE	1047
6.194.2.3 RHS_EXT	1047
6.194.2.4 RHS_TYPE	1048
6.194.3 Constructor & Destructor Documentation	1048
6.194.3.1 Divides_Vector_Vector()	1048
6.194.4 Member Function Documentation	1048
6.194.4.1 operator[]()	1048
6.194.4.2 size()	1048
6.194.5 Member Data Documentation	1049
6.194.5.1 lhs	1049
6.194.5.2 rhs	1049
6.195 Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > Class Template Reference	1049
6.195.1 Detailed Description	1050
6.195.2 Member Typedef Documentation	1050
6.195.2.1 LHS_EXT	1050
6.195.2.2 LHS_TYPE	1051
6.195.2.3 RHS_EXT	1051
6.195.2.4 RHS_TYPE	1051
6.195.2.5 STORAGE	1051
6.195.3 Constructor & Destructor Documentation	1051
6.195.3.1 Divides_Vector_Vector()	1052
6.195.4 Member Function Documentation	1052
6.195.4.1 operator[]()	1052
6.195.4.2 size()	1052



---

6.195.5 Member Data Documentation	1052
6.195.5.1 lhs	1052
6.195.5.2 rhs	1053
6.196 Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > Class Template Reference	1053
6.196.1 Detailed Description	1054
6.196.2 Member Typedef Documentation	1054
6.196.2.1 LHS_EXT	1054
6.196.2.2 LHS_TYPE	1054
6.196.2.3 RHS_EXT	1055
6.196.2.4 RHS_TYPE	1055
6.196.2.5 STORAGE	1055
6.196.3 Constructor & Destructor Documentation	1055
6.196.3.1 Divides_Vector_Vector()	1055
6.196.4 Member Function Documentation	1055
6.196.4.1 operator[]()	1056
6.196.4.2 size()	1056
6.196.5 Member Data Documentation	1056
6.196.5.1 lhs	1056
6.196.5.2 rhs	1056
6.197 Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > Class Template Reference	1057
6.197.1 Detailed Description	1058
6.197.2 Member Typedef Documentation	1058
6.197.2.1 LHS_EXT	1058
6.197.2.2 LHS_TYPE	1058
6.197.2.3 RHS_EXT	1058
6.197.2.4 RHS_TYPE	1059
6.197.2.5 STORAGE	1059
6.197.3 Constructor & Destructor Documentation	1059
6.197.3.1 Divides_Vector_Vector()	1059
6.197.4 Member Function Documentation	1059
6.197.4.1 operator[]()	1059
6.197.4.2 size()	1060
6.197.5 Member Data Documentation	1060
6.197.5.1 lhs	1060
6.197.5.2 rhs	1060
6.198 Rcpp::DottedPairImpl< CLASS > Class Template Reference	1060
6.198.1 Detailed Description	1061
6.198.2 Member Function Documentation	1061

---

6.198.2.1 insert()	1061
6.198.2.2 length()	1062
6.198.2.3 push_back()	1062
6.198.2.4 push_front()	1063
6.198.2.5 remove()	1064
6.198.2.6 replace()	1064
6.198.2.7 size()	1065
6.198.3 Friends And Related Function Documentation	1065
6.198.3.1 operator<<	1065
6.198.3.2 operator>>	1065
6.199 Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy Class Reference	1066
6.199.1 Detailed Description	1067
6.199.2 Constructor & Destructor Documentation	1067
6.199.2.1 DottedPairProxy()	1067
6.199.3 Member Function Documentation	1067
6.199.3.1 get()	1068
6.199.3.2 operator SEXP()	1068
6.199.3.3 operator T()	1068
6.199.3.4 operator=() [1/6]	1069
6.199.3.5 operator=() [2/6]	1069
6.199.3.6 operator=() [3/6]	1070
6.199.3.7 operator=() [4/6]	1070
6.199.3.8 operator=() [5/6]	1070
6.199.3.9 operator=() [6/6]	1071
6.199.3.10 set() [1/2]	1071
6.199.3.11 set() [2/2]	1071
6.199.4 Member Data Documentation	1072
6.199.4.1 node	1072
6.200 Rcpp::DottedPairProxyPolicy< CLASS > Class Template Reference	1072
6.200.1 Detailed Description	1072
6.200.2 Member Function Documentation	1073
6.200.2.1 operator[]() [1/2]	1073
6.200.2.2 operator[]() [2/2]	1073
6.201 Rcpp::internal::element_converter< RTYPE > Class Template Reference	1073
6.201.1 Detailed Description	1074
6.201.2 Member Typedef Documentation	1074
6.201.2.1 target	1074
6.201.3 Member Function Documentation	1074
6.201.3.1 get() [1/2]	1074

---

6.201.3.2 <code>get()</code> [2/2]	1075
6.202 <code>Rcpp::traits::enable_if&lt; B, T &gt;</code> Struct Template Reference	1075
6.202.1 Detailed Description	1075
6.203 <code>Rcpp::traits::enable_if&lt; true, T &gt;</code> Struct Template Reference	1075
6.203.1 Detailed Description	1075
6.203.2 Member Typedef Documentation	1076
6.203.2.1 <code>type</code>	1076
6.204 <code>Rcpp::enum_&lt; Enum, Parent &gt;</code> Class Template Reference	1076
6.204.1 Detailed Description	1077
6.204.2 Member Typedef Documentation	1077
6.204.2.1 <code>MAP</code>	1077
6.204.2.2 <code>PAIR</code>	1077
6.204.2.3 <code>self</code>	1077
6.204.3 Constructor & Destructor Documentation	1077
6.204.3.1 <code>enum_()</code>	1078
6.204.3.2 <code>~enum_()</code>	1078
6.204.4 Member Function Documentation	1078
6.204.4.1 <code>value()</code>	1078
6.204.5 Member Data Documentation	1079
6.204.5.1 <code>name</code>	1079
6.204.5.2 <code>parent_typeinfo_name</code>	1079
6.204.5.3 <code>values</code>	1079
6.205 <code>Rcpp::exception</code> Class Reference	1080
6.205.1 Detailed Description	1081
6.205.2 Constructor & Destructor Documentation	1081
6.205.2.1 <code>exception()</code> [1/2]	1081
6.205.2.2 <code>exception()</code> [2/2]	1082
6.205.2.3 <code>~exception()</code>	1082
6.205.3 Member Function Documentation	1082
6.205.3.1 <code>copy_stack_trace_to_r()</code>	1083
6.205.3.2 <code>include_call()</code>	1083
6.205.3.3 <code>record_stack_trace()</code>	1084
6.205.3.4 <code>what()</code>	1084
6.205.4 Member Data Documentation	1084
6.205.4.1 <code>include_call_</code>	1084
6.205.4.2 <code>message</code>	1084
6.205.4.3 <code>stack</code>	1085
6.206 <code>Rcpp::algorithm::helpers::exp</code> Struct Reference	1085
6.206.1 Detailed Description	1085

---

6.206.2 Member Function Documentation	1085
6.206.2.1 operator>()	1086
6.207 Rcpp::traits::expands_to_logical< T > Struct Template Reference	1086
6.207.1 Detailed Description	1087
6.208 Rcpp::traits::expands_to_logical__impl< RTYPE > Struct Template Reference	1088
6.208.1 Detailed Description	1089
6.209 Rcpp::traits::expands_to_logical__impl< LGLSXP > Struct Reference	1090
6.209.1 Detailed Description	1090
6.210 Rcpp::stats::ExpGenerator Class Reference	1090
6.210.1 Detailed Description	1091
6.210.2 Constructor & Destructor Documentation	1091
6.210.2.1 ExpGenerator()	1091
6.210.3 Member Function Documentation	1092
6.210.3.1 operator>()	1092
6.210.4 Member Data Documentation	1092
6.210.4.1 scale	1092
6.211 Rcpp::stats::ExpGenerator__rate1 Class Reference	1093
6.211.1 Detailed Description	1093
6.211.2 Constructor & Destructor Documentation	1094
6.211.2.1 ExpGenerator__rate1()	1094
6.211.3 Member Function Documentation	1094
6.211.3.1 operator>()	1094
6.212 Rcpp::attributes::ExportsGenerator Class Reference	1095
6.212.1 Detailed Description	1096
6.212.2 Constructor & Destructor Documentation	1096
6.212.2.1 ExportsGenerator() [1/2]	1096
6.212.2.2 ExportsGenerator() [2/2]	1097
6.212.2.3 ~ExportsGenerator()	1097
6.212.3 Member Function Documentation	1097
6.212.3.1 commit() [1/2]	1097
6.212.3.2 commit() [2/2]	1098
6.212.3.3 dotNameHelper()	1098
6.212.3.4 doWriteFunctions()	1098
6.212.3.5 exportValidationFunction()	1098
6.212.3.6 exportValidationFunctionRegisteredName()	1099
6.212.3.7 generatorToken()	1099
6.212.3.8 hasCppInterface()	1099
6.212.3.9 isSafeToOverwrite()	1100
6.212.3.10 operator std::ostream &()	1100

---

6.212.3.11 operator=()	1100
6.212.3.12 ostr()	1101
6.212.3.13 package()	1101
6.212.3.14 packageCpp()	1101
6.212.3.15 packageCppPrefix()	1102
6.212.3.16 registerCallableExportedName()	1102
6.212.3.17 remove()	1103
6.212.3.18 targetFile()	1103
6.212.3.19 writeBegin()	1103
6.212.3.20 writeEnd()	1104
6.212.3.21 writeFunctions()	1104
6.212.4 Member Data Documentation	1104
6.212.4.1 codeStream_	1104
6.212.4.2 commentPrefix_	1105
6.212.4.3 existingCode_	1105
6.212.4.4 hasCplusplusInterface_	1105
6.212.4.5 package_	1105
6.212.4.6 packageCpp_	1105
6.212.4.7 targetFile_	1106
6.213 Rcpp::attributes::ExportsGenerators Class Reference	1106
6.213.1 Detailed Description	1106
6.213.2 Member Typedef Documentation	1107
6.213.2.1 ltr	1107
6.213.3 Constructor & Destructor Documentation	1107
6.213.3.1 ExportsGenerators() [1/2]	1107
6.213.3.2 ~ExportsGenerators()	1107
6.213.3.3 ExportsGenerators() [2/2]	1107
6.213.4 Member Function Documentation	1107
6.213.4.1 add()	1108
6.213.4.2 commit()	1108
6.213.4.3 operator=()	1108
6.213.4.4 remove()	1108
6.213.4.5 writeBegin()	1109
6.213.4.6 writeEnd()	1109
6.213.4.7 writeFunctions()	1109
6.213.5 Member Data Documentation	1109
6.213.5.1 generators_	1109
6.214 Rcpp::traits::Extractor< RTYPE, NA, VECTOR > Struct Template Reference	1110
6.214.1 Detailed Description	1110

---

6.214.2 Member Typedef Documentation	1110
6.214.2.1 type	1110
6.215 tinyformat::detail::is_convertible< T1, T2 >::fail Struct Reference	1110
6.215.1 Detailed Description	1111
6.215.2 Member Data Documentation	1111
6.215.2.1 dummy	1111
6.216 Rcpp::Fast< VECTOR > Class Template Reference	1111
6.216.1 Detailed Description	1112
6.216.2 Member Typedef Documentation	1112
6.216.2.1 value_type	1112
6.216.3 Constructor & Destructor Documentation	1112
6.216.3.1 Fast()	1112
6.216.4 Member Function Documentation	1112
6.216.4.1 operator[]() [1/2]	1112
6.216.4.2 operator[]() [2/2]	1113
6.216.4.3 size()	1113
6.216.5 Member Data Documentation	1113
6.216.5.1 data	1113
6.216.5.2 v	1113
6.217 Rcpp::stats::FGenerator_Finite_Finite Class Reference	1114
6.217.1 Detailed Description	1115
6.217.2 Constructor & Destructor Documentation	1115
6.217.2.1 FGenerator_Finite_Finite()	1115
6.217.3 Member Function Documentation	1115
6.217.3.1 operator>()	1115
6.217.4 Member Data Documentation	1115
6.217.4.1 n1__2	1115
6.217.4.2 n2__2	1116
6.217.4.3 ratio	1116
6.218 Rcpp::stats::FGenerator_Finite_NotFinite Class Reference	1116
6.218.1 Detailed Description	1117
6.218.2 Constructor & Destructor Documentation	1117
6.218.2.1 FGenerator_Finite_NotFinite()	1117
6.218.3 Member Function Documentation	1118
6.218.3.1 operator>()	1118
6.218.4 Member Data Documentation	1118
6.218.4.1 n1	1118
6.218.4.2 n1__2	1118
6.219 Rcpp::stats::FGenerator_NotFinite_Finite Class Reference	1119

---

6.219.1 Detailed Description	1120
6.219.2 Constructor & Destructor Documentation	1120
6.219.2.1 FGenerator_NotFinite_Finite()	1120
6.219.3 Member Function Documentation	1120
6.219.3.1 operator>()	1120
6.219.4 Member Data Documentation	1120
6.219.4.1 n2	1120
6.219.4.2 n2__2	1121
6.220 Rcpp::FieldProxyPolicy< CLASS >::FieldProxy Class Reference	1121
6.220.1 Detailed Description	1122
6.220.2 Constructor & Destructor Documentation	1122
6.220.2.1 FieldProxy()	1122
6.220.3 Member Function Documentation	1123
6.220.3.1 get()	1123
6.220.3.2 operator SEXP()	1123
6.220.3.3 operator T()	1124
6.220.3.4 operator=() [1/4]	1124
6.220.3.5 operator=() [2/4]	1124
6.220.3.6 operator=() [3/4]	1125
6.220.3.7 operator=() [4/4]	1125
6.220.3.8 set()	1125
6.220.4 Member Data Documentation	1126
6.220.4.1 field_name	1126
6.220.4.2 parent	1126
6.221 Rcpp::FieldProxyPolicy< CLASS > Class Template Reference	1126
6.221.1 Detailed Description	1127
6.221.2 Member Function Documentation	1127
6.221.2.1 field() [1/2]	1127
6.221.2.2 field() [2/2]	1127
6.222 Rcpp::file_exists Class Reference	1128
6.222.1 Detailed Description	1129
6.222.2 Constructor & Destructor Documentation	1129
6.222.2.1 file_exists()	1129
6.223 Rcpp::file_io_error Class Reference	1129
6.223.1 Detailed Description	1130
6.223.2 Constructor & Destructor Documentation	1130
6.223.2.1 file_io_error() [1/3]	1130
6.223.2.2 file_io_error() [2/3]	1131
6.223.2.3 file_io_error() [3/3]	1131

6.223.2.4 ~file_io_error()	1131
6.223.3 Member Function Documentation	1131
6.223.3.1 filePath()	1131
6.223.3.2 what()	1132
6.223.4 Member Data Documentation	1132
6.223.4.1 file	1132
6.223.4.2 message	1132
6.224 Rcpp::file_not_found Class Reference	1133
6.224.1 Detailed Description	1134
6.224.2 Constructor & Destructor Documentation	1134
6.224.2.1 file_not_found()	1134
6.225 Rcpp::attributes::FileInfo Class Reference	1134
6.225.1 Detailed Description	1135
6.225.2 Constructor & Destructor Documentation	1135
6.225.2.1 FileInfo() [1/2]	1135
6.225.2.2 FileInfo() [2/2]	1135
6.225.3 Member Function Documentation	1135
6.225.3.1 exists()	1136
6.225.3.2 extension()	1136
6.225.3.3 lastModified()	1136
6.225.3.4 operator!=(())	1136
6.225.3.5 operator<()	1136
6.225.3.6 operator<<()	1137
6.225.3.7 operator==(())	1137
6.225.3.8 path()	1137
6.225.3.9 toList()	1137
6.225.4 Member Data Documentation	1137
6.225.4.1 exists_	1138
6.225.4.2 lastModified_	1138
6.225.4.3 path_	1138
6.226 Rcpp::fixed_call< RESULT_TYPE > Class Template Reference	1138
6.226.1 Detailed Description	1139
6.226.2 Member Typedef Documentation	1139
6.226.2.1 result_type	1139
6.226.3 Constructor & Destructor Documentation	1139
6.226.3.1 fixed_call() [1/2]	1139
6.226.3.2 fixed_call() [2/2]	1139
6.226.4 Member Function Documentation	1140
6.226.4.1 operator>()()	1140



---

6.226.5 Member Data Documentation	1140
6.226.5.1 call	1140
6.227 Rcpp::sugar::forbidden_conversion< bool > Class Template Reference	1140
6.227.1 Detailed Description	1140
6.228 Rcpp::sugar::forbidden_conversion< true > Class Reference	1141
6.228.1 Detailed Description	1141
6.229 tinyformat::detail::FormatArg Class Reference	1141
6.229.1 Detailed Description	1141
6.229.2 Constructor & Destructor Documentation	1142
6.229.2.1 FormatArg() [1/2]	1142
6.229.2.2 FormatArg() [2/2]	1142
6.229.3 Member Function Documentation	1142
6.229.3.1 format()	1142
6.229.3.2 formatImpl()	1143
6.229.3.3 toInt()	1143
6.229.3.4 toIntImpl()	1143
6.229.4 Member Data Documentation	1144
6.229.4.1 m_formatImpl	1144
6.229.4.2 m_toIntImpl	1144
6.229.4.3 m_value	1144
6.230 tinyformat::FormatList Class Reference	1145
6.230.1 Detailed Description	1146
6.230.2 Constructor & Destructor Documentation	1146
6.230.2.1 FormatList()	1146
6.230.3 Friends And Related Function Documentation	1146
6.230.3.1 vformat	1146
6.230.4 Member Data Documentation	1146
6.230.4.1 m_formatters	1147
6.230.4.2 m_N	1147
6.231 tinyformat::detail::FormatListN< N > Class Template Reference	1147
6.231.1 Detailed Description	1148
6.231.2 Member Function Documentation	1148
6.231.2.1 init()	1149
6.231.3 Member Data Documentation	1149
6.231.3.1 m_formatterStore	1149
6.232 tinyformat::detail::FormatListN< 0 > Class Reference	1149
6.232.1 Detailed Description	1150
6.232.2 Constructor & Destructor Documentation	1150
6.232.2.1 FormatListN()	1150

---

6.233 <a href="#">tinyformat::detail::formatValueAsType&lt; T, fmtT, convertible &gt; Struct Template Reference</a> . . . . .	1151
6.233.1 Detailed Description . . . . .	1151
6.233.2 Member Function Documentation . . . . .	1151
6.233.2.1 <a href="#">invoke()</a> . . . . .	1151
6.234 <a href="#">tinyformat::detail::formatValueAsType&lt; T, fmtT, true &gt; Struct Template Reference</a> . . . . .	1151
6.234.1 Detailed Description . . . . .	1152
6.234.2 Member Function Documentation . . . . .	1152
6.234.2.1 <a href="#">invoke()</a> . . . . .	1152
6.235 <a href="#">Rcpp::attributes::Function Class Reference</a> . . . . .	1152
6.235.1 Detailed Description . . . . .	1153
6.235.2 Constructor & Destructor Documentation . . . . .	1153
6.235.2.1 <a href="#">Function()</a> [1/2] . . . . .	1153
6.235.2.2 <a href="#">Function()</a> [2/2] . . . . .	1153
6.235.3 Member Function Documentation . . . . .	1154
6.235.3.1 <a href="#">arguments()</a> . . . . .	1154
6.235.3.2 <a href="#">empty()</a> . . . . .	1154
6.235.3.3 <a href="#">isHidden()</a> . . . . .	1155
6.235.3.4 <a href="#">name()</a> . . . . .	1155
6.235.3.5 <a href="#">operator"!=( )</a> . . . . .	1155
6.235.3.6 <a href="#">operator==( )</a> . . . . .	1156
6.235.3.7 <a href="#">renamedTo()</a> . . . . .	1156
6.235.3.8 <a href="#">signature()</a> [1/2] . . . . .	1157
6.235.3.9 <a href="#">signature()</a> [2/2] . . . . .	1157
6.235.3.10 <a href="#">type()</a> . . . . .	1158
6.235.4 Member Data Documentation . . . . .	1158
6.235.4.1 <a href="#">arguments_</a> . . . . .	1158
6.235.4.2 <a href="#">name_</a> . . . . .	1158
6.235.4.3 <a href="#">type_</a> . . . . .	1158
6.236 <a href="#">Rcpp::FunctionFinalizer&lt; Class &gt; Class Template Reference</a> . . . . .	1159
6.236.1 Detailed Description . . . . .	1160
6.236.2 Member Typedef Documentation . . . . .	1160
6.236.2.1 <a href="#">Pointer</a> . . . . .	1160
6.236.3 Constructor & Destructor Documentation . . . . .	1160
6.236.3.1 <a href="#">FunctionFinalizer()</a> . . . . .	1160
6.236.4 Member Function Documentation . . . . .	1160
6.236.4.1 <a href="#">run()</a> . . . . .	1161
6.236.5 Member Data Documentation . . . . .	1161
6.236.5.1 <a href="#">finalizer</a> . . . . .	1161
6.237 <a href="#">Rcpp::stats::GammaGenerator Class Reference</a> . . . . .	1161

6.237.1 Detailed Description	1162
6.237.2 Constructor & Destructor Documentation	1162
6.237.2.1 GammaGenerator()	1162
6.237.3 Member Function Documentation	1163
6.237.3.1 operator>()	1163
6.237.4 Member Data Documentation	1163
6.237.4.1 a	1163
6.237.4.2 scale	1163
6.238 Rcpp::Generator< T > Class Template Reference	1163
6.238.1 Detailed Description	1164
6.238.2 Member Typedef Documentation	1164
6.238.2.1 r_generator	1164
6.239 Rcpp::internal::generic_element_converter< RTYPE > Class Template Reference	1164
6.239.1 Detailed Description	1165
6.239.2 Member Typedef Documentation	1165
6.239.2.1 target	1165
6.239.3 Member Function Documentation	1165
6.239.3.1 get() [1/3]	1165
6.239.3.2 get() [2/3]	1166
6.239.3.3 get() [3/3]	1166
6.240 Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy > Class Template Reference	1167
6.240.1 Detailed Description	1168
6.240.2 Member Typedef Documentation	1168
6.240.2.1 VECTOR	1168
6.240.3 Constructor & Destructor Documentation	1168
6.240.3.1 generic_name_proxy() [1/2]	1168
6.240.3.2 generic_name_proxy() [2/2]	1168
6.240.3.3 ~generic_name_proxy()	1169
6.240.4 Member Function Documentation	1169
6.240.4.1 get()	1169
6.240.4.2 operator SEXP()	1169
6.240.4.3 operator T()	1170
6.240.4.4 operator=() [1/3]	1170
6.240.4.5 operator=() [2/3]	1171
6.240.4.6 operator=() [3/3]	1171
6.240.4.7 set()	1172
6.240.5 Member Data Documentation	1172
6.240.5.1 name	1172
6.240.5.2 parent	1173

---

6.241 Rcpp::internal::generic_proxy< RTYPE, StoragePolicy > Class Template Reference	1173
6.241.1 Detailed Description	1175
6.241.2 Member Typedef Documentation	1175
6.241.2.1 VECTOR	1175
6.241.3 Constructor & Destructor Documentation	1175
6.241.3.1 generic_proxy() [1/3]	1175
6.241.3.2 generic_proxy() [2/3]	1175
6.241.3.3 generic_proxy() [3/3]	1176
6.241.4 Member Function Documentation	1176
6.241.4.1 get()	1176
6.241.4.2 import()	1176
6.241.4.3 move()	1177
6.241.4.4 operator bool()	1177
6.241.4.5 operator int()	1177
6.241.4.6 operator SEXP()	1178
6.241.4.7 operator U()	1178
6.241.4.8 operator=() [1/4]	1179
6.241.4.9 operator=() [2/4]	1180
6.241.4.10 operator=() [3/4]	1180
6.241.4.11 operator=() [4/4]	1181
6.241.4.12 set()	1181
6.241.4.13 swap()	1182
6.241.5 Member Data Documentation	1182
6.241.5.1 index	1182
6.241.5.2 parent	1183
6.242 Rcpp::GenericProxy< Proxy > Struct Template Reference	1183
6.242.1 Detailed Description	1183
6.242.2 Member Function Documentation	1183
6.242.2.1 get()	1183
6.243 Rcpp::stats::GeomGenerator Class Reference	1184
6.243.1 Detailed Description	1185
6.243.2 Constructor & Destructor Documentation	1185
6.243.2.1 GeomGenerator()	1185
6.243.3 Member Function Documentation	1185
6.243.3.1 operator>()	1185
6.243.4 Member Data Documentation	1186
6.243.4.1 lambda	1186
6.244 Rcpp::sugar::Grabber< HASH, RTYPE > Class Template Reference	1186
6.244.1 Detailed Description	1187

6.244.2 Constructor & Destructor Documentation	1187
6.244.2.1 Grabber()	1187
6.244.3 Member Function Documentation	1187
6.244.3.1 operator>()	1187
6.244.4 Member Data Documentation	1188
6.244.4.1 index	1188
6.244.4.2 names	1188
6.244.4.3 res	1188
6.245 Rcpp::traits::has_iterator< T > Struct Template Reference	1189
6.245.1 Detailed Description	1189
6.246 Rcpp::traits::has_na< RTYPE > Struct Template Reference	1190
6.246.1 Detailed Description	1191
6.247 Rcpp::traits::has_na< CPLXSP > Struct Reference	1191
6.247.1 Detailed Description	1192
6.248 Rcpp::traits::has_na< INTSP > Struct Reference	1192
6.248.1 Detailed Description	1193
6.249 Rcpp::traits::has_na< LGLSP > Struct Reference	1193
6.249.1 Detailed Description	1194
6.250 Rcpp::traits::has_na< REALSP > Struct Reference	1194
6.250.1 Detailed Description	1195
6.251 Rcpp::traits::has_na< STRSP > Struct Reference	1195
6.251.1 Detailed Description	1196
6.252 Rcpp::sugar::cbind_impl::detail::has_stored_type< T > Class Template Reference	1196
6.252.1 Detailed Description	1196
6.252.2 Member Typedef Documentation	1197
6.252.2.1 yes	1197
6.252.3 Member Function Documentation	1197
6.252.3.1 test() [1/2]	1197
6.252.3.2 test() [2/2]	1197
6.252.4 Member Data Documentation	1197
6.252.4.1 value	1197
6.253 Rcpp::sugar::Head< RTYPE, NA, T > Class Template Reference	1198
6.253.1 Detailed Description	1199
6.253.2 Member Typedef Documentation	1199
6.253.2.1 STORAGE	1199
6.253.2.2 VEC_TYPE	1199
6.253.3 Constructor & Destructor Documentation	1199
6.253.3.1 Head()	1199
6.253.4 Member Function Documentation	1200

---

6.253.4.1 operator[]()	1200
6.253.4.2 size()	1200
6.253.5 Member Data Documentation	1200
6.253.5.1 n	1200
6.253.5.2 object	1200
6.254 Rcpp::stats::HyperGenerator Class Reference	1201
6.254.1 Detailed Description	1202
6.254.2 Constructor & Destructor Documentation	1202
6.254.2.1 HyperGenerator()	1202
6.254.3 Member Function Documentation	1202
6.254.3.1 operator()()	1202
6.254.4 Member Data Documentation	1202
6.254.4.1 kk	1202
6.254.4.2 nn1	1203
6.254.4.3 nn2	1203
6.255 Rcpp::traits::identity< T > Struct Template Reference	1203
6.255.1 Detailed Description	1203
6.255.2 Member Typedef Documentation	1203
6.255.2.1 type	1204
6.256 Rcpp::traits::if_< COND, LHS, RHS > Struct Template Reference	1204
6.256.1 Detailed Description	1204
6.256.2 Member Typedef Documentation	1204
6.256.2.1 type	1204
6.257 Rcpp::traits::if_< false, LHS, RHS > Struct Template Reference	1205
6.257.1 Detailed Description	1205
6.257.2 Member Typedef Documentation	1205
6.257.2.1 type	1205
6.258 Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	1206
6.258.1 Detailed Description	1207
6.258.2 Member Typedef Documentation	1207
6.258.2.1 COND_TYPE	1207
6.258.2.2 LHS_TYPE	1207
6.258.2.3 RHS_TYPE	1207
6.258.2.4 STORAGE	1208
6.258.3 Constructor & Destructor Documentation	1208
6.258.3.1 IfElse()	1208
6.258.4 Member Function Documentation	1208
6.258.4.1 operator[]()	1208

---

6.258.4.2 size()	1209
6.258.5 Member Data Documentation	1209
6.258.5.1 cond	1210
6.258.5.2 lhs	1210
6.258.5.3 rhs	1210
6.259 Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	1211
6.259.1 Detailed Description	1212
6.259.2 Member Typedef Documentation	1212
6.259.2.1 COND_TYPE	1212
6.259.2.2 LHS_EXT	1212
6.259.2.3 LHS_TYPE	1212
6.259.2.4 RHS_EXT	1213
6.259.2.5 RHS_TYPE	1213
6.259.2.6 STORAGE	1213
6.259.3 Constructor & Destructor Documentation	1213
6.259.3.1 IfElse()	1213
6.259.4 Member Function Documentation	1213
6.259.4.1 operator[]()	1214
6.259.4.2 size()	1214
6.259.5 Member Data Documentation	1214
6.259.5.1 cond	1214
6.259.5.2 lhs	1215
6.259.5.3 rhs	1215
6.260 Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T > Class Template Reference	1215
6.260.1 Detailed Description	1216
6.260.2 Member Typedef Documentation	1216
6.260.2.1 COND_TYPE	1216
6.260.2.2 STORAGE	1216
6.260.3 Constructor & Destructor Documentation	1217
6.260.3.1 IfElse_Primitive_Primitive()	1217
6.260.4 Member Function Documentation	1217
6.260.4.1 operator[]()	1217
6.260.4.2 size()	1218
6.260.5 Member Data Documentation	1218
6.260.5.1 cond	1218
6.260.5.2 lhs	1218
6.260.5.3 na	1219
6.260.5.4 rhs	1219

---

6.261 Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T > Class Template Reference . . . . .	1219
6.261.1 Detailed Description . . . . .	1220
6.261.2 Member Typedef Documentation . . . . .	1220
6.261.2.1 COND_TYPE . . . . .	1220
6.261.2.2 STORAGE . . . . .	1220
6.261.3 Constructor & Destructor Documentation . . . . .	1221
6.261.3.1 IfElse_Primitive_Primitive() . . . . .	1221
6.261.4 Member Function Documentation . . . . .	1221
6.261.4.1 operator[]() . . . . .	1221
6.261.4.2 size() . . . . .	1221
6.261.5 Member Data Documentation . . . . .	1222
6.261.5.1 cond . . . . .	1222
6.261.5.2 lhs . . . . .	1222
6.261.5.3 rhs . . . . .	1222
6.262 Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T > Class Template Reference . . . . .	1223
6.262.1 Detailed Description . . . . .	1224
6.262.2 Member Typedef Documentation . . . . .	1224
6.262.2.1 COND_TYPE . . . . .	1224
6.262.2.2 RHS_EXT . . . . .	1224
6.262.2.3 RHS_TYPE . . . . .	1224
6.262.2.4 STORAGE . . . . .	1225
6.262.3 Constructor & Destructor Documentation . . . . .	1225
6.262.3.1 IfElse_Primitive_Vector() . . . . .	1225
6.262.4 Member Function Documentation . . . . .	1225
6.262.4.1 operator[]() . . . . .	1225
6.262.4.2 size() . . . . .	1226
6.262.5 Member Data Documentation . . . . .	1226
6.262.5.1 cond . . . . .	1226
6.262.5.2 lhs . . . . .	1227
6.262.5.3 rhs . . . . .	1227
6.263 Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T > Class Template Reference . . . . .	1227
6.263.1 Detailed Description . . . . .	1228
6.263.2 Member Typedef Documentation . . . . .	1228
6.263.2.1 COND_TYPE . . . . .	1228
6.263.2.2 RHS_EXT . . . . .	1229
6.263.2.3 RHS_TYPE . . . . .	1229
6.263.2.4 STORAGE . . . . .	1229
6.263.3 Constructor & Destructor Documentation . . . . .	1229



---

6.263.3.1	IfElse_Primitive_Vector()	1229
6.263.4	Member Function Documentation	1229
6.263.4.1	operator[]()	1230
6.263.4.2	size()	1230
6.263.5	Member Data Documentation	1230
6.263.5.1	cond	1230
6.263.5.2	lhs	1231
6.263.5.3	rhs	1231
6.264	Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T > Class Template Reference	1231
6.264.1	Detailed Description	1232
6.264.2	Member Typedef Documentation	1232
6.264.2.1	COND_TYPE	1232
6.264.2.2	LHS_EXT	1233
6.264.2.3	LHS_TYPE	1233
6.264.2.4	STORAGE	1233
6.264.3	Constructor & Destructor Documentation	1233
6.264.3.1	IfElse_Vector_Primitive()	1233
6.264.4	Member Function Documentation	1233
6.264.4.1	operator[]()	1234
6.264.4.2	size()	1234
6.264.5	Member Data Documentation	1235
6.264.5.1	cond	1235
6.264.5.2	lhs	1235
6.264.5.3	rhs	1235
6.265	Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T > Class Template Reference	1236
6.265.1	Detailed Description	1237
6.265.2	Member Typedef Documentation	1237
6.265.2.1	COND_TYPE	1237
6.265.2.2	LHS_EXT	1237
6.265.2.3	LHS_TYPE	1237
6.265.2.4	STORAGE	1238
6.265.3	Constructor & Destructor Documentation	1238
6.265.3.1	IfElse_Vector_Primitive()	1238
6.265.4	Member Function Documentation	1238
6.265.4.1	operator[]()	1238
6.265.4.2	size()	1239
6.265.5	Member Data Documentation	1239
6.265.5.1	cond	1239

---

6.265.5.2 lhs	1239
6.265.5.3 rhs	1240
6.266 Rcpp::sugar::In< RTYPE, TABLE_T > Class Template Reference	1240
6.266.1 Detailed Description	1240
6.266.2 Member Typedef Documentation	1240
6.266.2.1 HASH	1241
6.266.3 Constructor & Destructor Documentation	1241
6.266.3.1 In()	1241
6.266.4 Member Function Documentation	1241
6.266.4.1 get()	1241
6.266.5 Member Data Documentation	1242
6.266.5.1 hash	1242
6.266.5.2 vec	1242
6.267 Rcpp::sugar::IndexHash< RTYPE > Class Template Reference	1242
6.267.1 Detailed Description	1243
6.267.2 Member Typedef Documentation	1243
6.267.2.1 STORAGE	1244
6.267.2.2 VECTOR	1244
6.267.3 Constructor & Destructor Documentation	1244
6.267.3.1 IndexHash()	1244
6.267.4 Member Function Documentation	1244
6.267.4.1 add_value()	1245
6.267.4.2 contains()	1245
6.267.4.3 fill()	1246
6.267.4.4 fill_and_get_duplicated()	1246
6.267.4.5 get_addr() [1/4]	1247
6.267.4.6 get_addr() [2/4]	1247
6.267.4.7 get_addr() [3/4]	1247
6.267.4.8 get_addr() [4/4]	1247
6.267.4.9 get_index()	1248
6.267.4.10 get_profile_data()	1248
6.267.4.11 keys()	1249
6.267.4.12 lookup() [1/2]	1249
6.267.4.13 lookup() [2/2]	1250
6.267.4.14 lookup__impl()	1250
6.267.4.15 not_equal()	1251
6.267.4.16 size()	1251
6.267.5 Member Data Documentation	1251
6.267.5.1 data	1251

---

6.267.5.2 k	1251
6.267.5.3 m	1252
6.267.5.4 n	1252
6.267.5.5 size_	1252
6.267.5.6 src	1252
6.268 Rcpp::traits::init_type< RTYPE > Struct Template Reference	1253
6.268.1 Detailed Description	1253
6.268.2 Member Typedef Documentation	1253
6.268.2.1 type	1253
6.269 Rcpp::traits::init_type< LGLSXP > Struct Reference	1253
6.269.1 Detailed Description	1254
6.269.2 Member Typedef Documentation	1254
6.269.2.1 type	1254
6.270 Rcpp::traits::init_type< STRSXP > Struct Reference	1254
6.270.1 Detailed Description	1254
6.270.2 Member Typedef Documentation	1254
6.270.2.1 type	1254
6.271 Rcpp::traits::input_parameter< T > Struct Template Reference	1255
6.271.1 Detailed Description	1255
6.271.2 Member Typedef Documentation	1255
6.271.2.1 type	1255
6.272 Rcpp::traits::input_parameter< const T & > Struct Template Reference	1255
6.272.1 Detailed Description	1256
6.272.2 Member Typedef Documentation	1256
6.272.2.1 type	1256
6.273 Rcpp::traits::input_parameter< const T > Struct Template Reference	1256
6.273.1 Detailed Description	1256
6.273.2 Member Typedef Documentation	1256
6.273.2.1 type	1257
6.274 Rcpp::traits::input_parameter< T & > Struct Template Reference	1257
6.274.1 Detailed Description	1257
6.274.2 Member Typedef Documentation	1257
6.274.2.1 type	1257
6.275 Rcpp::InputParameter< T > Class Template Reference	1258
6.275.1 Detailed Description	1258
6.275.2 Constructor & Destructor Documentation	1258
6.275.2.1 InputParameter()	1258
6.275.3 Member Function Documentation	1258
6.275.3.1 operator T()	1258

---

6.275.4 Member Data Documentation	1259
6.275.4.1 x	1259
6.276 Rcpp::sugar::InSet< HASH > Class Template Reference	1259
6.276.1 Detailed Description	1259
6.276.2 Member Typedef Documentation	1260
6.276.2.1 STORAGE	1260
6.276.3 Constructor & Destructor Documentation	1260
6.276.3.1 InSet()	1260
6.276.4 Member Function Documentation	1260
6.276.4.1 operator>()	1260
6.276.5 Member Data Documentation	1260
6.276.5.1 hash	1261
6.277 Rcpp::traits::int2type< I > Struct Template Reference	1261
6.277.1 Detailed Description	1261
6.277.2 Member Enumeration Documentation	1261
6.277.2.1 anonymous enum	1261
6.278 Rcpp::traits::integral_constant< _T, _V > Struct Template Reference	1262
6.278.1 Detailed Description	1262
6.278.2 Member Typedef Documentation	1263
6.278.2.1 type	1263
6.278.2.2 value_type	1263
6.278.3 Member Data Documentation	1263
6.278.3.1 value	1263
6.279 interrupt_exception Class Reference	1264
6.279.1 Detailed Description	1264
6.279.2 Constructor & Destructor Documentation	1265
6.279.2.1 interrupt_exception()	1265
6.279.2.2 ~interrupt_exception()	1265
6.279.3 Member Function Documentation	1265
6.279.3.1 what()	1265
6.279.4 Member Data Documentation	1266
6.279.4.1 detailed_message	1266
6.280 Rcpp::internal::InterruptedException Class Reference	1266
6.280.1 Detailed Description	1266
6.281 Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	1266
6.281.1 Detailed Description	1267
6.281.2 Member Typedef Documentation	1267
6.281.2.1 ITERATOR	1267
6.281.2.2 SET	1267

---

6.281.2.3 STORAGE	1268
6.281.3 Constructor & Destructor Documentation	1268
6.281.3.1 Intersect()	1268
6.281.4 Member Function Documentation	1268
6.281.4.1 get()	1269
6.281.5 Member Data Documentation	1269
6.281.5.1 intersect	1269
6.282 Rcpp::traits::is_arithmetic< typename > Struct Template Reference	1270
6.282.1 Detailed Description	1270
6.283 Rcpp::traits::is_arithmetic< const double > Struct Reference	1271
6.283.1 Detailed Description	1271
6.284 Rcpp::traits::is_arithmetic< const float > Struct Reference	1272
6.284.1 Detailed Description	1272
6.285 Rcpp::traits::is_arithmetic< const int > Struct Reference	1273
6.285.1 Detailed Description	1273
6.286 Rcpp::traits::is_arithmetic< const long > Struct Reference	1274
6.286.1 Detailed Description	1274
6.287 Rcpp::traits::is_arithmetic< const long double > Struct Reference	1275
6.287.1 Detailed Description	1275
6.288 Rcpp::traits::is_arithmetic< const short > Struct Reference	1276
6.288.1 Detailed Description	1276
6.289 Rcpp::traits::is_arithmetic< const unsigned int > Struct Reference	1277
6.289.1 Detailed Description	1277
6.290 Rcpp::traits::is_arithmetic< const unsigned long > Struct Reference	1278
6.290.1 Detailed Description	1278
6.291 Rcpp::traits::is_arithmetic< const unsigned short > Struct Reference	1279
6.291.1 Detailed Description	1279
6.292 Rcpp::traits::is_arithmetic< double > Struct Reference	1280
6.292.1 Detailed Description	1280
6.293 Rcpp::traits::is_arithmetic< float > Struct Reference	1281
6.293.1 Detailed Description	1281
6.294 Rcpp::traits::is_arithmetic< int > Struct Reference	1282
6.294.1 Detailed Description	1282
6.295 Rcpp::traits::is_arithmetic< long > Struct Reference	1283
6.295.1 Detailed Description	1283
6.296 Rcpp::traits::is_arithmetic< long double > Struct Reference	1284
6.296.1 Detailed Description	1284
6.297 Rcpp::traits::is_arithmetic< short > Struct Reference	1285
6.297.1 Detailed Description	1285

---

6.298 Rcpp::traits::is_arithmetic< unsigned int > Struct Reference	1286
6.298.1 Detailed Description	1286
6.299 Rcpp::traits::is_arithmetic< unsigned long > Struct Reference	1287
6.299.1 Detailed Description	1287
6.300 Rcpp::traits::is_arithmetic< unsigned short > Struct Reference	1288
6.300.1 Detailed Description	1288
6.301 Rcpp::traits::is_bool< typename > Struct Template Reference	1289
6.301.1 Detailed Description	1289
6.302 Rcpp::traits::is_bool< bool > Struct Reference	1290
6.302.1 Detailed Description	1290
6.303 Rcpp::traits::is_bool< const bool > Struct Reference	1291
6.303.1 Detailed Description	1291
6.304 Rcpp::traits::is_bool< volatile bool > Struct Reference	1292
6.304.1 Detailed Description	1292
6.305 Rcpp::traits::is_const< typename > Struct Template Reference	1293
6.305.1 Detailed Description	1293
6.306 Rcpp::traits::is_const< _Tp const > Struct Template Reference	1294
6.306.1 Detailed Description	1294
6.307 Rcpp::traits::is_convertible< T, U > Class Template Reference	1295
6.307.1 Detailed Description	1295
6.307.2 Member Typedef Documentation	1295
6.307.2.1 Small	1296
6.307.2.2 type	1296
6.307.2.3 value_type	1296
6.307.3 Member Function Documentation	1296
6.307.3.1 MakeT()	1296
6.307.3.2 Test() [1/2]	1296
6.307.3.3 Test() [2/2]	1297
6.307.4 Member Data Documentation	1297
6.307.4.1 value	1297
6.308 tinyformat::detail::is_convertible< T1, T2 > Struct Template Reference	1297
6.308.1 Detailed Description	1298
6.308.2 Member Function Documentation	1298
6.308.2.1 makeT1()	1298
6.308.2.2 tryConvert() [1/2]	1298
6.308.2.3 tryConvert() [2/2]	1298
6.308.3 Member Data Documentation	1298
6.308.3.1 value	1299
6.309 Rcpp::traits::is_eigen_base< T > Struct Template Reference	1299

---

6.309.1 Detailed Description	1300
6.310 Rcpp::traits::is_exporter< T > Struct Template Reference	1300
6.310.1 Detailed Description	1301
6.311 Rcpp::traits::is_generator< T > Struct Template Reference	1302
6.311.1 Detailed Description	1302
6.312 Rcpp::traits::is_importer< T > Struct Template Reference	1303
6.312.1 Detailed Description	1303
6.313 Rcpp::traits::is_module_object< T > Struct Template Reference	1304
6.313.1 Detailed Description	1305
6.314 Rcpp::traits::is_named< T > Struct Template Reference	1305
6.314.1 Detailed Description	1306
6.315 Rcpp::traits::is_named< named_object< T > > Struct Template Reference	1306
6.315.1 Detailed Description	1307
6.316 Rcpp::traits::is_named< Rcpp::Argument > Struct Reference	1307
6.316.1 Detailed Description	1308
6.317 Rcpp::traits::is_pointer< T > Struct Template Reference	1308
6.317.1 Detailed Description	1309
6.318 Rcpp::traits::is_pointer< T * > Struct Template Reference	1309
6.318.1 Detailed Description	1310
6.319 Rcpp::traits::is_primitive< T > Struct Template Reference	1310
6.319.1 Detailed Description	1311
6.320 Rcpp::traits::is_reference< typename > Struct Template Reference	1311
6.320.1 Detailed Description	1312
6.321 Rcpp::traits::is_reference< _Tp & > Struct Template Reference	1313
6.321.1 Detailed Description	1313
6.322 Rcpp::traits::is_sugar_expression< T > Struct Template Reference	1314
6.322.1 Detailed Description	1314
6.323 Rcpp::sugar::is_sugar_vector< T > Struct Template Reference	1315
6.323.1 Detailed Description	1316
6.324 Rcpp::sugar::is_sugar_vector< Rcpp::Vector< RTYPE > > Struct Template Reference	1316
6.324.1 Detailed Description	1317
6.325 Rcpp::traits::is_trivial< RTYPE > Struct Template Reference	1317
6.325.1 Detailed Description	1318
6.326 Rcpp::traits::is_trivial< EXPRXP > Struct Reference	1318
6.326.1 Detailed Description	1319
6.327 Rcpp::traits::is_trivial< VECSXP > Struct Reference	1319
6.327.1 Detailed Description	1320
6.328 tinyformat::detail::is_wchar< T > Struct Template Reference	1320
6.328.1 Detailed Description	1320

6.328.2 Member Typedef Documentation	1321
6.328.2.1 tinyformat_wchar_is_not_supported	1321
6.329 tinyformat::detail::is_wchar< const wchar_t * > Struct Reference	1321
6.329.1 Detailed Description	1321
6.330 tinyformat::detail::is_wchar< const wchar_t[n]> Struct Template Reference	1321
6.330.1 Detailed Description	1321
6.331 tinyformat::detail::is_wchar< wchar_t * > Struct Reference	1322
6.331.1 Detailed Description	1322
6.332 tinyformat::detail::is_wchar< wchar_t[n]> Struct Template Reference	1322
6.332.1 Detailed Description	1322
6.333 Rcpp::traits::is_wide_string< T > Struct Template Reference	1322
6.333.1 Detailed Description	1323
6.334 Rcpp::traits::is_wide_string< char > Struct Reference	1324
6.334.1 Detailed Description	1325
6.335 Rcpp::traits::is_wide_string< const char * > Struct Reference	1325
6.335.1 Detailed Description	1326
6.336 Rcpp::traits::is_wide_string< const wchar_t * > Struct Reference	1326
6.336.1 Detailed Description	1327
6.337 Rcpp::traits::is_wide_string< wchar_t > Struct Reference	1327
6.337.1 Detailed Description	1328
6.338 Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE > Class Template Reference	1328
6.338.1 Detailed Description	1329
6.338.2 Constructor & Destructor Documentation	1329
6.338.2.1 IsFinite()	1330
6.338.3 Member Function Documentation	1330
6.338.3.1 operator[]()	1330
6.338.3.2 size()	1330
6.338.4 Member Data Documentation	1330
6.338.4.1 obj	1330
6.339 Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE > Class Template Reference	1331
6.339.1 Detailed Description	1332
6.339.2 Constructor & Destructor Documentation	1332
6.339.2.1 IsInfinite()	1333
6.339.3 Member Function Documentation	1333
6.339.3.1 operator[]()	1333
6.339.3.2 size()	1333
6.339.4 Member Data Documentation	1333
6.339.4.1 obj	1333
6.340 Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE > Class Template Reference	1334



---

6.340.1 Detailed Description	1335
6.340.2 Member Typedef Documentation	1335
6.340.2.1 BASE	1335
6.340.2.2 STORAGE	1335
6.340.3 Constructor & Destructor Documentation	1335
6.340.3.1 IsNa()	1336
6.340.4 Member Function Documentation	1336
6.340.4.1 operator[]()	1336
6.340.4.2 size()	1336
6.340.5 Member Data Documentation	1337
6.340.5.1 obj	1337
6.341 Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE > Class Template Reference	1337
6.341.1 Detailed Description	1338
6.341.2 Member Typedef Documentation	1338
6.341.2.1 BASE	1338
6.341.2.2 STORAGE	1339
6.341.3 Constructor & Destructor Documentation	1339
6.341.3.1 IsNa()	1339
6.341.4 Member Function Documentation	1339
6.341.4.1 operator[]()	1339
6.341.4.2 size()	1339
6.341.5 Member Data Documentation	1340
6.341.5.1 obj	1340
6.342 Rcpp::sugar::IsNa_Vector_is_na< T > Class Template Reference	1340
6.342.1 Detailed Description	1341
6.342.2 Constructor & Destructor Documentation	1341
6.342.2.1 IsNa_Vector_is_na()	1342
6.342.3 Member Function Documentation	1342
6.342.3.1 operator[]()	1342
6.342.3.2 size()	1342
6.342.4 Member Data Documentation	1342
6.342.4.1 ref	1342
6.343 Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE > Class Template Reference	1343
6.343.1 Detailed Description	1344
6.343.2 Constructor & Destructor Documentation	1344
6.343.2.1 IsNaN()	1345
6.343.3 Member Function Documentation	1345
6.343.3.1 operator[]()	1345
6.343.3.2 size()	1345

---

6.343.4 Member Data Documentation	1345
6.343.4.1 obj	1345
6.344 Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS > Class Template Reference	1346
6.344.1 Detailed Description	1347
6.344.2 Member Typedef Documentation	1347
6.344.2.1 difference_type	1347
6.344.2.2 iterator_category	1347
6.344.2.3 pointer	1347
6.344.2.4 reference	1348
6.344.2.5 value_type	1348
6.344.2.6 vector_iterator	1348
6.344.3 Constructor & Destructor Documentation	1348
6.344.3.1 iter_base() [1/2]	1348
6.344.3.2 iter_base() [2/2]	1349
6.344.4 Member Function Documentation	1349
6.344.4.1 operator!=(())	1349
6.344.4.2 operator*(())	1349
6.344.4.3 operator+()	1350
6.344.4.4 operator++() [1/2]	1350
6.344.4.5 operator++() [2/2]	1350
6.344.4.6 operator+=(())	1351
6.344.4.7 operator-() [1/3]	1351
6.344.4.8 operator-() [2/3]	1351
6.344.4.9 operator-() [3/3]	1352
6.344.4.10 operator--() [1/2]	1352
6.344.4.11 operator--() [2/2]	1352
6.344.4.12 operator==(())	1353
6.344.4.13 operator->()	1353
6.344.4.14 operator<()	1353
6.344.4.15 operator<=(())	1353
6.344.4.16 operator==(())	1354
6.344.4.17 operator>()	1354
6.344.4.18 operator>=(())	1354
6.344.4.19 operator[]()	1354
6.344.5 Member Data Documentation	1355
6.344.5.1 index	1355
6.344.5.2 row	1355
6.345 Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS > Class Template Reference	1355
6.345.1 Detailed Description	1356

---

6.345.2 Member Typedef Documentation	1356
6.345.2.1 difference_type	1357
6.345.2.2 iterator_category	1357
6.345.2.3 pointer	1357
6.345.2.4 reference	1357
6.345.2.5 value_type	1357
6.345.3 Constructor & Destructor Documentation	1358
6.345.3.1 iter_base()	1358
6.345.4 Member Function Documentation	1358
6.345.4.1 operator!=(())	1358
6.345.4.2 operator*()	1358
6.345.4.3 operator+()	1359
6.345.4.4 operator++() [1/2]	1359
6.345.4.5 operator++() [2/2]	1359
6.345.4.6 operator+=(())	1359
6.345.4.7 operator-() [1/2]	1360
6.345.4.8 operator-() [2/2]	1360
6.345.4.9 operator--() [1/2]	1360
6.345.4.10 operator--() [2/2]	1360
6.345.4.11 operator==(())	1361
6.345.4.12 operator->()	1361
6.345.4.13 operator<()	1361
6.345.4.14 operator<=()	1361
6.345.4.15 operator==(())	1362
6.345.4.16 operator>()	1362
6.345.4.17 operator>=()	1362
6.345.4.18 operator[]()	1362
6.345.5 Member Data Documentation	1363
6.345.5.1 index	1363
6.345.5.2 object	1363
6.346 Rcpp::MatrixRow< RTYPE >::iter_traits Struct Reference	1363
6.346.1 Detailed Description	1364
6.346.2 Member Typedef Documentation	1364
6.346.2.1 difference_type	1364
6.346.2.2 pointer	1364
6.346.2.3 reference	1364
6.346.2.4 value_type	1364
6.346.2.5 vector_iterator	1365
6.347 Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits Struct Reference	1365

6.347.1 Detailed Description	1365
6.347.2 Member Typedef Documentation	1365
6.347.2.1 difference_type	1365
6.347.2.2 iterator_category	1366
6.347.2.3 pointer	1366
6.347.2.4 reference	1366
6.347.2.5 value_type	1366
6.348 Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator Class Reference	1367
6.348.1 Detailed Description	1368
6.348.2 Member Typedef Documentation	1368
6.348.2.1 difference_type	1368
6.348.2.2 iterator_category	1369
6.348.2.3 pointer	1369
6.348.2.4 reference	1369
6.348.2.5 value_type	1369
6.348.3 Constructor & Destructor Documentation	1369
6.348.3.1 iterator() [1/2]	1370
6.348.3.2 iterator() [2/2]	1370
6.348.4 Member Function Documentation	1370
6.348.4.1 index()	1370
6.348.4.2 operator!=(())	1371
6.348.4.3 operator*()	1371
6.348.4.4 operator+()	1371
6.348.4.5 operator++() [1/2]	1372
6.348.4.6 operator++() [2/2]	1372
6.348.4.7 operator+=(())	1372
6.348.4.8 operator-() [1/2]	1373
6.348.4.9 operator-() [2/2]	1373
6.348.4.10 operator--() [1/2]	1374
6.348.4.11 operator--() [2/2]	1374
6.348.4.12 operator-=()	1374
6.348.4.13 operator->()	1375
6.348.4.14 operator<()	1375
6.348.4.15 operator<=()	1375
6.348.4.16 operator==(())	1376
6.348.4.17 operator>()	1376
6.348.4.18 operator>=()	1377
6.348.4.19 update_index()	1377
6.348.5 Member Data Documentation	1378

---

6.348.5.1 i	1378
6.348.5.2 j	1378
6.348.5.3 nc	1378
6.348.5.4 nr	1379
6.348.5.5 object	1379
6.349 Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 > Class Template Reference	1379
6.349.1 Detailed Description	1380
6.349.2 Member Typedef Documentation	1380
6.349.2.1 stored_type	1381
6.349.3 Constructor & Destructor Documentation	1381
6.349.3.1 JoinOp()	1381
6.349.4 Member Function Documentation	1381
6.349.4.1 ncol()	1381
6.349.4.2 nrow()	1382
6.349.4.3 operator>()	1382
6.349.4.4 operator[]()	1382
6.349.4.5 size()	1383
6.349.5 Member Data Documentation	1383
6.349.5.1 e1	1383
6.349.5.2 e2	1383
6.350 Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > > Class Template Reference	1384
6.350.1 Detailed Description	1385
6.350.2 Member Typedef Documentation	1385
6.350.2.1 E2	1385
6.350.2.2 stored_type	1386
6.350.3 Constructor & Destructor Documentation	1386
6.350.3.1 JoinOp()	1386
6.350.4 Member Function Documentation	1386
6.350.4.1 ncol()	1386
6.350.4.2 nrow()	1386
6.350.4.3 operator>()	1387
6.350.4.4 operator[]()	1387
6.350.4.5 size()	1387
6.350.5 Member Data Documentation	1388
6.350.5.1 e1	1388
6.350.5.2 e2	1388
6.351 Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 > Class Template Reference	1388
6.351.1 Detailed Description	1389

---

6.351.2 Member Typedef Documentation	1390
6.351.2.1 E1	1390
6.351.2.2 stored_type	1390
6.351.3 Constructor & Destructor Documentation	1390
6.351.3.1 JoinOp()	1390
6.351.4 Member Function Documentation	1390
6.351.4.1 ncol()	1391
6.351.4.2 nrow()	1391
6.351.4.3 operator>()	1391
6.351.4.4 operator[]()	1392
6.351.4.5 size()	1392
6.351.5 Member Data Documentation	1392
6.351.5.1 e1	1392
6.351.5.2 e2	1392
6.352 Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > > Class Template Reference	1393
6.352.1 Detailed Description	1394
6.352.2 Member Typedef Documentation	1394
6.352.2.1 E1	1394
6.352.2.2 E2	1394
6.352.2.3 stored_type	1395
6.352.3 Constructor & Destructor Documentation	1395
6.352.3.1 JoinOp()	1395
6.352.4 Member Function Documentation	1395
6.352.4.1 ncol()	1395
6.352.4.2 nrow()	1395
6.352.4.3 operator>()	1396
6.352.4.4 operator[]()	1396
6.352.4.5 size()	1396
6.352.5 Member Data Documentation	1397
6.352.5.1 e1	1397
6.352.5.2 e2	1397
6.353 Rcpp::sugar::Lapply< RTYPE, NA, T, Function > Class Template Reference	1397
6.353.1 Detailed Description	1399
6.353.2 Member Typedef Documentation	1399
6.353.2.1 result_type	1399
6.353.2.2 VEC	1399
6.353.3 Constructor & Destructor Documentation	1399
6.353.3.1 Lapply()	1399

---

6.353.4 Member Function Documentation	1400
6.353.4.1 operator[]()	1400
6.353.4.2 size()	1400
6.353.5 Member Data Documentation	1401
6.353.5.1 fun	1401
6.353.5.2 vec	1401
6.354 Rcpp::sugar::Lazy< T, EXPR > Class Template Reference	1401
6.354.1 Detailed Description	1402
6.354.2 Member Function Documentation	1402
6.354.2.1 operator T()	1402
6.355 Rcpp::internal::LazyVector< VECTOR > Class Template Reference	1402
6.355.1 Detailed Description	1403
6.355.2 Member Typedef Documentation	1403
6.355.2.1 r_type	1403
6.355.2.2 stored_type	1403
6.355.3 Constructor & Destructor Documentation	1403
6.355.3.1 LazyVector()	1404
6.355.4 Member Function Documentation	1404
6.355.4.1 operator[]()	1404
6.355.5 Member Data Documentation	1404
6.355.5.1 data	1404
6.355.5.2 known	1404
6.355.5.3 n	1405
6.355.5.4 vec	1405
6.356 Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > > Class Template Reference	1405
6.356.1 Detailed Description	1406
6.356.2 Member Typedef Documentation	1406
6.356.2.1 Proxy	1406
6.356.2.2 VECTOR	1406
6.356.3 Constructor & Destructor Documentation	1407
6.356.3.1 LazyVector()	1407
6.356.4 Member Function Documentation	1407
6.356.4.1 operator[]()	1407
6.356.5 Member Data Documentation	1407
6.356.5.1 vec	1407
6.357 Rcpp::ListOf< T > Class Template Reference	1408
6.357.1 Detailed Description	1409
6.357.2 Member Typedef Documentation	1409
6.357.2.1 const_iterator	1409

---

6.357.2.2 iterator	1410
6.357.3 Constructor & Destructor Documentation	1410
6.357.3.1 ListOf() [1/4]	1410
6.357.3.2 ListOf() [2/4]	1410
6.357.3.3 ListOf() [3/4]	1411
6.357.3.4 ListOf() [4/4]	1411
6.357.4 Member Function Documentation	1411
6.357.4.1 begin() [1/2]	1412
6.357.4.2 begin() [2/2]	1412
6.357.4.3 end() [1/2]	1413
6.357.4.4 end() [2/2]	1413
6.357.4.5 get()	1413
6.357.4.6 operator List()	1414
6.357.4.7 operator SEXP()	1414
6.357.4.8 operator=() [1/2]	1414
6.357.4.9 operator=() [2/2]	1415
6.357.4.10 operator[]() [1/4]	1415
6.357.4.11 operator[]() [2/4]	1415
6.357.4.12 operator[]() [3/4]	1416
6.357.4.13 operator[]() [4/4]	1416
6.357.4.14 size()	1416
6.357.5 Member Data Documentation	1416
6.357.5.1 list	1417
6.358 Rcpp::stats::LNormGenerator Class Reference	1417
6.358.1 Detailed Description	1418
6.358.2 Constructor & Destructor Documentation	1418
6.358.2.1 LNormGenerator()	1418
6.358.3 Member Function Documentation	1418
6.358.3.1 operator()()	1418
6.358.4 Member Data Documentation	1419
6.358.4.1 meanlog	1419
6.358.4.2 sdlog	1419
6.359 Rcpp::stats::LNormGenerator_0 Class Reference	1419
6.359.1 Detailed Description	1420
6.359.2 Constructor & Destructor Documentation	1420
6.359.2.1 LNormGenerator_0()	1420
6.359.3 Member Function Documentation	1420
6.359.3.1 operator()()	1421
6.360 Rcpp::stats::LNormGenerator_1 Class Reference	1421



6.360.1 Detailed Description	1422
6.360.2 Constructor & Destructor Documentation	1422
6.360.2.1 LNormGenerator_1()	1422
6.360.3 Member Function Documentation	1423
6.360.3.1 operator()()	1423
6.360.4 Member Data Documentation	1423
6.360.4.1 meanlog	1423
6.361 Rcpp::algorithm::helpers::log Struct Reference	1423
6.361.1 Detailed Description	1424
6.361.2 Member Function Documentation	1424
6.361.2.1 operator()()	1424
6.362 Rcpp::stats::LogisGenerator Class Reference	1425
6.362.1 Detailed Description	1426
6.362.2 Constructor & Destructor Documentation	1426
6.362.2.1 LogisGenerator()	1426
6.362.3 Member Function Documentation	1426
6.362.3.1 operator()()	1426
6.362.4 Member Data Documentation	1427
6.362.4.1 location	1427
6.362.4.2 scale	1427
6.363 Rcpp::stats::LogisGenerator_0 Class Reference	1427
6.363.1 Detailed Description	1428
6.363.2 Constructor & Destructor Documentation	1428
6.363.2.1 LogisGenerator_0()	1428
6.363.3 Member Function Documentation	1428
6.363.3.1 operator()()	1429
6.364 Rcpp::stats::LogisGenerator_1 Class Reference	1429
6.364.1 Detailed Description	1430
6.364.2 Constructor & Destructor Documentation	1430
6.364.2.1 LogisGenerator_1()	1430
6.364.3 Member Function Documentation	1431
6.364.3.1 operator()()	1431
6.364.4 Member Data Documentation	1431
6.364.4.1 location	1431
6.365 Rcpp::LongjumpException Struct Reference	1431
6.365.1 Detailed Description	1432
6.365.2 Constructor & Destructor Documentation	1432
6.365.2.1 LongjumpException()	1432
6.365.3 Member Data Documentation	1432

---

6.365.3.1 token	1433
6.366 Rcpp::sugar::LowerTri< RTYPE, NA, T > Class Template Reference	1433
6.366.1 Detailed Description	1435
6.366.2 Member Typedef Documentation	1435
6.366.2.1 MatBase	1435
6.366.2.2 Method	1435
6.366.3 Constructor & Destructor Documentation	1435
6.366.3.1 LowerTri()	1436
6.366.4 Member Function Documentation	1436
6.366.4.1 get()	1436
6.366.4.2 get_diag_false()	1436
6.366.4.3 get_diag_true()	1436
6.366.4.4 ncol()	1437
6.366.4.5 nrow()	1437
6.366.4.6 operator>()	1437
6.366.4.7 size()	1438
6.366.5 Member Data Documentation	1438
6.366.5.1 getter	1438
6.366.5.2 nc	1438
6.366.5.3 nr	1438
6.367 Rcpp::lsinfo Struct Reference	1439
6.367.1 Detailed Description	1439
6.367.2 Member Data Documentation	1439
6.367.2.1 ls_corr	1439
6.367.2.2 ls_trans	1439
6.368 Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, is_container > Struct Template Reference	1439
6.368.1 Detailed Description	1440
6.369 Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true > Struct Reference	1440
6.369.1 Detailed Description	1440
6.369.2 Member Typedef Documentation	1440
6.369.2.1 stored_type	1440
6.369.2.2 T	1441
6.369.3 Member Enumeration Documentation	1441
6.369.3.1 anonymous enum	1441
6.369.4 Member Function Documentation	1441
6.369.4.1 operator>()	1441
6.370 Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true > Struct Reference	1442
6.370.1 Detailed Description	1442
6.370.2 Member Typedef Documentation	1442

---

6.370.2.1 stored_type . . . . .	1442
6.370.2.2 T . . . . .	1442
6.370.3 Member Enumeration Documentation . . . . .	1442
6.370.3.1 anonymous enum . . . . .	1442
6.370.4 Member Function Documentation . . . . .	1443
6.370.4.1 operator>() . . . . .	1443
6.371 Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, false > Struct Template Reference . . . . .	1443
6.371.1 Detailed Description . . . . .	1444
6.371.2 Member Enumeration Documentation . . . . .	1444
6.371.2.1 anonymous enum . . . . .	1444
6.371.3 Member Function Documentation . . . . .	1444
6.371.3.1 operator>() . . . . .	1444
6.372 Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, true > Struct Template Reference . . . . .	1445
6.372.1 Detailed Description . . . . .	1445
6.372.2 Member Typedef Documentation . . . . .	1445
6.372.2.1 stored_type . . . . .	1446
6.372.3 Member Enumeration Documentation . . . . .	1446
6.372.3.1 anonymous enum . . . . .	1446
6.372.4 Member Function Documentation . . . . .	1446
6.372.4.1 operator>() . . . . .	1446
6.373 Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function > Class Template Reference . . . . .	1447
6.373.1 Detailed Description . . . . .	1448
6.373.2 Member Typedef Documentation . . . . .	1448
6.373.2.1 result_type . . . . .	1448
6.373.3 Constructor & Destructor Documentation . . . . .	1448
6.373.3.1 Mapply_2() . . . . .	1448
6.373.4 Member Function Documentation . . . . .	1448
6.373.4.1 operator[]() . . . . .	1449
6.373.4.2 size() . . . . .	1449
6.373.5 Member Data Documentation . . . . .	1449
6.373.5.1 fun . . . . .	1449
6.373.5.2 vec_1 . . . . .	1449
6.373.5.3 vec_2 . . . . .	1450
6.374 Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function > Class Template Reference . . . . .	1450
6.374.1 Detailed Description . . . . .	1451
6.374.2 Member Typedef Documentation . . . . .	1451
6.374.2.1 result_type . . . . .	1451
6.374.3 Constructor & Destructor Documentation . . . . .	1451

6.374.3.1 Mapply_2_Primitive_Vector()	1451
6.374.4 Member Function Documentation	1452
6.374.4.1 operator[]()	1452
6.374.4.2 size()	1452
6.374.5 Member Data Documentation	1452
6.374.5.1 fun	1452
6.374.5.2 prim_1	1453
6.374.5.3 vec_2	1453
6.375 Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function > Class Template Reference	1453
6.375.1 Detailed Description	1454
6.375.2 Member Typedef Documentation	1454
6.375.2.1 result_type	1454
6.375.3 Constructor & Destructor Documentation	1454
6.375.3.1 Mapply_2_Vector_Primitive()	1455
6.375.4 Member Function Documentation	1455
6.375.4.1 operator[]()	1455
6.375.4.2 size()	1455
6.375.5 Member Data Documentation	1455
6.375.5.1 fun	1456
6.375.5.2 prim_2	1456
6.375.5.3 vec_1	1456
6.376 Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function > Class Template Reference	1457
6.376.1 Detailed Description	1458
6.376.2 Member Typedef Documentation	1458
6.376.2.1 EXT_1	1458
6.376.2.2 EXT_2	1458
6.376.2.3 EXT_3	1459
6.376.2.4 result_type	1459
6.376.2.5 VEC_1	1459
6.376.2.6 VEC_2	1459
6.376.2.7 VEC_3	1460
6.376.3 Constructor & Destructor Documentation	1460
6.376.3.1 Mapply_3()	1460
6.376.4 Member Function Documentation	1460
6.376.4.1 operator[]()	1460
6.376.4.2 size()	1461
6.376.5 Member Data Documentation	1461
6.376.5.1 fun	1461

---

6.376.5.2	vec_1	1461
6.376.5.3	vec_2	1462
6.376.5.4	vec_3	1462
6.377	Rcpp::Matrix< RTYPE, StoragePolicy > Class Template Reference	1462
6.377.1	Detailed Description	1465
6.377.2	Member Typedef Documentation	1465
6.377.2.1	Column	1465
6.377.2.2	const_iterator	1465
6.377.2.3	const_Proxy	1465
6.377.2.4	ConstColumn	1466
6.377.2.5	ConstRow	1466
6.377.2.6	converter_type	1466
6.377.2.7	iterator	1466
6.377.2.8	Proxy	1466
6.377.2.9	Row	1467
6.377.2.10	Storage	1467
6.377.2.11	stored_type	1467
6.377.2.12	Sub	1467
6.377.2.13	VECTOR	1467
6.377.3	Constructor & Destructor Documentation	1467
6.377.3.1	Matrix() [1/10]	1468
6.377.3.2	Matrix() [2/10]	1468
6.377.3.3	Matrix() [3/10]	1468
6.377.3.4	Matrix() [4/10]	1469
6.377.3.5	Matrix() [5/10]	1469
6.377.3.6	Matrix() [6/10]	1469
6.377.3.7	Matrix() [7/10]	1470
6.377.3.8	Matrix() [8/10]	1470
6.377.3.9	Matrix() [9/10]	1470
6.377.3.10	Matrix() [10/10]	1471
6.377.4	Member Function Documentation	1471
6.377.4.1	at() [1/2]	1471
6.377.4.2	at() [2/2]	1472
6.377.4.3	begin() [1/2]	1472
6.377.4.4	begin() [2/2]	1472
6.377.4.5	cbegin()	1473
6.377.4.6	cend()	1473
6.377.4.7	cols()	1474
6.377.4.8	column() [1/2]	1474

---

6.377.4.9 column() [2/2]	1474
6.377.4.10 diag()	1475
6.377.4.11 end() [1/2]	1475
6.377.4.12 end() [2/2]	1476
6.377.4.13 fill_diag()	1476
6.377.4.14 fill_diag_dispatch() [1/2]	1477
6.377.4.15 fill_diag_dispatch() [2/2]	1477
6.377.4.16 import_matrix_expression()	1478
6.377.4.17 ncol()	1479
6.377.4.18 nrow()	1479
6.377.4.19 offset()	1479
6.377.4.20 operator() [1/9]	1480
6.377.4.21 operator() [2/9]	1480
6.377.4.22 operator() [3/9]	1480
6.377.4.23 operator() [4/9]	1481
6.377.4.24 operator() [5/9]	1481
6.377.4.25 operator() [6/9]	1482
6.377.4.26 operator() [7/9]	1482
6.377.4.27 operator() [8/9]	1482
6.377.4.28 operator() [9/9]	1483
6.377.4.29 operator=() [1/2]	1483
6.377.4.30 operator=() [2/2]	1483
6.377.4.31 operator[]() [1/2]	1484
6.377.4.32 operator[]() [2/2]	1484
6.377.4.33 row() [1/2]	1484
6.377.4.34 row() [2/2]	1485
6.377.4.35 rows()	1485
6.377.5 Member Data Documentation	1485
6.377.5.1 nrows	1485
6.378 Rcpp::traits::matrix_interface< T > Struct Template Reference	1486
6.378.1 Detailed Description	1486
6.379 Rcpp::sugar::cbind_impl::detail::matrix_return< T, is_container > Struct Template Reference	1487
6.379.1 Detailed Description	1487
6.380 Rcpp::sugar::cbind_impl::matrix_return< T, B > Struct Template Reference	1487
6.380.1 Detailed Description	1488
6.381 Rcpp::sugar::cbind_impl::detail::matrix_return< bool, false > Struct Reference	1488
6.381.1 Detailed Description	1488
6.381.2 Member Typedef Documentation	1488
6.381.2.1 type	1488

6.382 Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Matrix< LGLSXP >, true > Struct Reference	1489
6.382.1 Detailed Description	1489
6.382.2 Member Typedef Documentation	1489
6.382.2.1 type	1489
6.383 Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Vector< LGLSXP >, true > Struct Reference	1489
6.383.1 Detailed Description	1489
6.383.2 Member Typedef Documentation	1490
6.383.2.1 type	1490
6.384 Rcpp::sugar::cbind_impl::detail::matrix_return< T, false > Struct Template Reference	1490
6.384.1 Detailed Description	1491
6.384.2 Member Typedef Documentation	1491
6.384.2.1 type	1491
6.384.3 Member Enumeration Documentation	1491
6.384.3.1 anonymous enum	1491
6.385 Rcpp::sugar::cbind_impl::matrix_return< T, false > Struct Template Reference	1491
6.385.1 Detailed Description	1492
6.386 Rcpp::sugar::cbind_impl::detail::matrix_return< T, true > Struct Template Reference	1493
6.386.1 Detailed Description	1493
6.386.2 Member Typedef Documentation	1493
6.386.2.1 stored_type	1493
6.386.2.2 type	1493
6.386.3 Member Enumeration Documentation	1493
6.386.3.1 anonymous enum	1493
6.387 Rcpp::MatrixBase< RTYPE, na, MATRIX > Class Template Reference	1494
6.387.1 Detailed Description	1495
6.387.2 Member Typedef Documentation	1495
6.387.2.1 stored_type	1495
6.387.3 Member Function Documentation	1496
6.387.3.1 begin()	1496
6.387.3.2 end()	1496
6.387.3.3 eye()	1497
6.387.3.4 get_ref()	1497
6.387.3.5 ncol()	1497
6.387.3.6 nrow()	1498
6.387.3.7 ones()	1498
6.387.3.8 operator>()()	1498
6.387.3.9 size()	1498
6.387.3.10 zeros()	1499
6.388 Rcpp::MatrixColumn< RTYPE > Class Template Reference	1499

---

6.388.1 Detailed Description	1501
6.388.2 Member Typedef Documentation	1501
6.388.2.1 const_iterator	1501
6.388.2.2 const_Proxy	1501
6.388.2.3 iterator	1501
6.388.2.4 MATRIX	1502
6.388.2.5 Proxy	1502
6.388.2.6 value_type	1502
6.388.3 Constructor & Destructor Documentation	1502
6.388.3.1 MatrixColumn() [1/3]	1502
6.388.3.2 MatrixColumn() [2/3]	1503
6.388.3.3 MatrixColumn() [3/3]	1503
6.388.4 Member Function Documentation	1503
6.388.4.1 begin() [1/2]	1503
6.388.4.2 begin() [2/2]	1504
6.388.4.3 cbegin()	1504
6.388.4.4 cend()	1504
6.388.4.5 end() [1/2]	1504
6.388.4.6 end() [2/2]	1505
6.388.4.7 operator=() [1/2]	1505
6.388.4.8 operator=() [2/2]	1505
6.388.4.9 operator[]() [1/2]	1506
6.388.4.10 operator[]() [2/2]	1506
6.388.4.11 size()	1506
6.388.5 Member Data Documentation	1506
6.388.5.1 const_start	1506
6.388.5.2 n	1507
6.388.5.3 start	1507
6.389 Rcpp::MatrixRow< RTYPE > Class Template Reference	1507
6.389.1 Detailed Description	1509
6.389.2 Member Typedef Documentation	1509
6.389.2.1 const_iterator	1509
6.389.2.2 const_reference	1509
6.389.2.3 iterator	1510
6.389.2.4 MATRIX	1510
6.389.2.5 Proxy	1510
6.389.2.6 reference	1510
6.389.2.7 value_type	1510
6.389.3 Constructor & Destructor Documentation	1510



---

6.389.3.1 MatrixRow() [1/2]	1511
6.389.3.2 MatrixRow() [2/2]	1511
6.389.4 Member Function Documentation	1511
6.389.4.1 begin() [1/2]	1511
6.389.4.2 begin() [2/2]	1512
6.389.4.3 cbegin()	1512
6.389.4.4 cend()	1512
6.389.4.5 end() [1/2]	1512
6.389.4.6 end() [2/2]	1513
6.389.4.7 get_parent_index()	1513
6.389.4.8 operator=() [1/2]	1513
6.389.4.9 operator=() [2/2]	1514
6.389.4.10 operator[]() [1/2]	1514
6.389.4.11 operator[]() [2/2]	1515
6.389.4.12 size()	1515
6.389.5 Member Data Documentation	1515
6.389.5.1 parent	1515
6.389.5.2 parent_nrow	1516
6.389.5.3 row	1516
6.389.5.4 start	1516
6.390 Rcpp::sugar::Max< RTYPE, NA, T > Class Template Reference	1516
6.390.1 Detailed Description	1517
6.390.2 Member Typedef Documentation	1517
6.390.2.1 STORAGE	1517
6.390.3 Constructor & Destructor Documentation	1517
6.390.3.1 Max()	1517
6.390.4 Member Function Documentation	1517
6.390.4.1 operator STORAGE()	1518
6.390.5 Member Data Documentation	1518
6.390.5.1 obj	1518
6.391 Rcpp::sugar::Max< RTYPE, false, T > Class Template Reference	1518
6.391.1 Detailed Description	1519
6.391.2 Member Typedef Documentation	1519
6.391.2.1 STORAGE	1519
6.391.3 Constructor & Destructor Documentation	1519
6.391.3.1 Max()	1519
6.391.4 Member Function Documentation	1519
6.391.4.1 operator STORAGE()	1520
6.391.5 Member Data Documentation	1520

---

6.391.5.1 obj	1520
6.392 Rcpp::sugar::Mean< RTYPE, NA, T > Class Template Reference	1520
6.392.1 Detailed Description	1522
6.392.2 Member Typedef Documentation	1522
6.392.2.1 VEC_TYPE	1522
6.392.2.2 VECTOR	1522
6.392.3 Constructor & Destructor Documentation	1522
6.392.3.1 Mean()	1522
6.392.4 Member Function Documentation	1522
6.392.4.1 get()	1523
6.392.5 Member Data Documentation	1523
6.392.5.1 object	1523
6.393 Rcpp::sugar::Mean< CPLXSP, NA, T > Class Template Reference	1524
6.393.1 Detailed Description	1525
6.393.2 Member Typedef Documentation	1525
6.393.2.1 VEC_TYPE	1525
6.393.3 Constructor & Destructor Documentation	1525
6.393.3.1 Mean()	1525
6.393.4 Member Function Documentation	1525
6.393.4.1 get()	1526
6.393.5 Member Data Documentation	1526
6.393.5.1 object	1526
6.394 Rcpp::sugar::Mean< INTSP, NA, T > Class Template Reference	1527
6.394.1 Detailed Description	1528
6.394.2 Member Typedef Documentation	1528
6.394.2.1 VEC_TYPE	1528
6.394.3 Constructor & Destructor Documentation	1528
6.394.3.1 Mean()	1528
6.394.4 Member Function Documentation	1528
6.394.4.1 get()	1529
6.394.5 Member Data Documentation	1529
6.394.5.1 object	1529
6.395 Rcpp::sugar::Mean< LGLSP, NA, T > Class Template Reference	1530
6.395.1 Detailed Description	1531
6.395.2 Member Typedef Documentation	1531
6.395.2.1 VEC_TYPE	1531
6.395.3 Constructor & Destructor Documentation	1531
6.395.3.1 Mean()	1531
6.395.4 Member Function Documentation	1531

---

6.395.4.1 get() . . . . .	1532
6.395.5 Member Data Documentation . . . . .	1532
6.395.5.1 object . . . . .	1532
6.396 Rcpp::sugar::Median< RTYPE, NA, T, NA_RM > Class Template Reference . . . . .	1532
6.396.1 Detailed Description . . . . .	1533
6.396.2 Member Typedef Documentation . . . . .	1533
6.396.2.1 result_type . . . . .	1533
6.396.2.2 stored_type . . . . .	1533
6.396.2.3 VECTOR . . . . .	1534
6.396.3 Member Enumeration Documentation . . . . .	1534
6.396.3.1 anonymous enum . . . . .	1534
6.396.4 Constructor & Destructor Documentation . . . . .	1534
6.396.4.1 Median() . . . . .	1534
6.396.5 Member Function Documentation . . . . .	1534
6.396.5.1 operator result_type() . . . . .	1535
6.396.6 Member Data Documentation . . . . .	1535
6.396.6.1 x . . . . .	1535
6.397 Rcpp::sugar::Median< RTYPE, false, T, NA_RM > Class Template Reference . . . . .	1536
6.397.1 Detailed Description . . . . .	1536
6.397.2 Member Typedef Documentation . . . . .	1536
6.397.2.1 result_type . . . . .	1536
6.397.2.2 stored_type . . . . .	1537
6.397.2.3 VECTOR . . . . .	1537
6.397.3 Member Enumeration Documentation . . . . .	1537
6.397.3.1 anonymous enum . . . . .	1537
6.397.4 Constructor & Destructor Documentation . . . . .	1537
6.397.4.1 Median() . . . . .	1537
6.397.5 Member Function Documentation . . . . .	1538
6.397.5.1 operator result_type() . . . . .	1538
6.397.6 Member Data Documentation . . . . .	1538
6.397.6.1 x . . . . .	1538
6.398 Rcpp::sugar::Median< RTYPE, NA, T, true > Class Template Reference . . . . .	1538
6.398.1 Detailed Description . . . . .	1539
6.398.2 Member Typedef Documentation . . . . .	1539
6.398.2.1 result_type . . . . .	1539
6.398.2.2 stored_type . . . . .	1539
6.398.2.3 VECTOR . . . . .	1540
6.398.3 Member Enumeration Documentation . . . . .	1540
6.398.3.1 anonymous enum . . . . .	1540

---

6.398.4 Constructor & Destructor Documentation	1540
6.398.4.1 Median()	1540
6.398.5 Member Function Documentation	1540
6.398.5.1 operator result_type()	1541
6.398.6 Member Data Documentation	1541
6.398.6.1 x	1541
6.399 Rcpp::sugar::Median< STRSXP, false, T, true > Class Template Reference	1541
6.399.1 Detailed Description	1542
6.399.2 Member Typedef Documentation	1542
6.399.2.1 result_type	1542
6.399.2.2 stored_type	1542
6.399.2.3 VECTOR	1543
6.399.3 Constructor & Destructor Documentation	1543
6.399.3.1 Median()	1543
6.399.4 Member Function Documentation	1543
6.399.4.1 operator result_type()	1543
6.399.5 Member Data Documentation	1544
6.399.5.1 x	1544
6.400 Rcpp::sugar::Median< STRSXP, NA, T, NA_RM > Class Template Reference	1544
6.400.1 Detailed Description	1544
6.400.2 Member Typedef Documentation	1545
6.400.2.1 result_type	1545
6.400.2.2 stored_type	1545
6.400.2.3 VECTOR	1545
6.400.3 Constructor & Destructor Documentation	1545
6.400.3.1 Median()	1545
6.400.4 Member Function Documentation	1546
6.400.4.1 operator result_type()	1546
6.400.5 Member Data Documentation	1546
6.400.5.1 x	1546
6.401 Rcpp::sugar::Median< STRSXP, NA, T, true > Class Template Reference	1547
6.401.1 Detailed Description	1547
6.401.2 Member Typedef Documentation	1547
6.401.2.1 result_type	1547
6.401.2.2 stored_type	1548
6.401.2.3 VECTOR	1548
6.401.3 Constructor & Destructor Documentation	1548
6.401.3.1 Median()	1548
6.401.4 Member Function Documentation	1548

---

6.401.4.1 operator result_type()	1548
6.401.5 Member Data Documentation	1549
6.401.5.1 x	1549
6.402 Rcpp::sugar::Min< RTYPE, NA, T > Class Template Reference	1549
6.402.1 Detailed Description	1549
6.402.2 Member Typedef Documentation	1550
6.402.2.1 STORAGE	1550
6.402.3 Constructor & Destructor Documentation	1550
6.402.3.1 Min()	1550
6.402.4 Member Function Documentation	1550
6.402.4.1 operator STORAGE()	1550
6.402.5 Member Data Documentation	1551
6.402.5.1 obj	1551
6.403 Rcpp::sugar::Min< RTYPE, false, T > Class Template Reference	1551
6.403.1 Detailed Description	1551
6.403.2 Member Typedef Documentation	1552
6.403.2.1 STORAGE	1552
6.403.3 Constructor & Destructor Documentation	1552
6.403.3.1 Min()	1552
6.403.4 Member Function Documentation	1552
6.403.4.1 operator STORAGE()	1552
6.403.5 Member Data Documentation	1553
6.403.5.1 obj	1553
6.404 Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T > Class Template Reference	1553
6.404.1 Detailed Description	1554
6.404.2 Member Typedef Documentation	1555
6.404.2.1 STORAGE	1555
6.404.2.2 VEC_EXT	1555
6.404.2.3 VEC_TYPE	1555
6.404.3 Constructor & Destructor Documentation	1555
6.404.3.1 Minus_Primitive_Vector()	1555
6.404.4 Member Function Documentation	1556
6.404.4.1 operator[]()	1556
6.404.4.2 size()	1556
6.404.5 Member Data Documentation	1556
6.404.5.1 lhs	1556
6.404.5.2 lhs_na	1557
6.404.5.3 rhs	1557
6.405 Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T > Class Template Reference	1557

---

6.405.1 Detailed Description	1559
6.405.2 Member Typedef Documentation	1559
6.405.2.1 VEC_EXT	1559
6.405.2.2 VEC_TYPE	1559
6.405.3 Constructor & Destructor Documentation	1560
6.405.3.1 Minus_Primitive_Vector()	1560
6.405.4 Member Function Documentation	1560
6.405.4.1 operator[]()	1560
6.405.4.2 size()	1560
6.405.5 Member Data Documentation	1560
6.405.5.1 lhs	1561
6.405.5.2 rhs	1561
6.406 Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T > Class Template Reference	1561
6.406.1 Detailed Description	1562
6.406.2 Member Typedef Documentation	1563
6.406.2.1 VEC_EXT	1563
6.406.2.2 VEC_TYPE	1563
6.406.3 Constructor & Destructor Documentation	1563
6.406.3.1 Minus_Primitive_Vector()	1563
6.406.4 Member Function Documentation	1563
6.406.4.1 operator[]()	1564
6.406.4.2 size()	1564
6.406.5 Member Data Documentation	1564
6.406.5.1 lhs	1564
6.406.5.2 rhs	1564
6.407 Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T > Class Template Reference	1565
6.407.1 Detailed Description	1566
6.407.2 Member Typedef Documentation	1567
6.407.2.1 STORAGE	1567
6.407.2.2 VEC_EXT	1567
6.407.2.3 VEC_TYPE	1567
6.407.3 Constructor & Destructor Documentation	1567
6.407.3.1 Minus_Primitive_Vector()	1567
6.407.4 Member Function Documentation	1568
6.407.4.1 operator[]()	1568
6.407.4.2 size()	1568
6.407.5 Member Data Documentation	1568
6.407.5.1 lhs	1568
6.407.5.2 lhs_na	1568

---

6.407.5.3 rhs	1569
6.408 Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T > Class Template Reference	1569
6.408.1 Detailed Description	1570
6.408.2 Member Typedef Documentation	1571
6.408.2.1 STORAGE	1571
6.408.2.2 VEC_EXT	1571
6.408.2.3 VEC_TYPE	1571
6.408.3 Constructor & Destructor Documentation	1571
6.408.3.1 Minus_Vector_Primitive()	1571
6.408.4 Member Function Documentation	1572
6.408.4.1 operator[]()	1572
6.408.4.2 size()	1572
6.408.5 Member Data Documentation	1572
6.408.5.1 lhs	1572
6.408.5.2 rhs	1573
6.408.5.3 rhs_na	1573
6.409 Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T > Class Template Reference	1573
6.409.1 Detailed Description	1575
6.409.2 Member Typedef Documentation	1575
6.409.2.1 VEC_EXT	1575
6.409.2.2 VEC_TYPE	1575
6.409.3 Constructor & Destructor Documentation	1576
6.409.3.1 Minus_Vector_Primitive()	1576
6.409.4 Member Function Documentation	1576
6.409.4.1 operator[]()	1576
6.409.4.2 size()	1576
6.409.5 Member Data Documentation	1576
6.409.5.1 lhs	1577
6.409.5.2 rhs	1577
6.410 Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T > Class Template Reference	1577
6.410.1 Detailed Description	1578
6.410.2 Member Typedef Documentation	1579
6.410.2.1 VEC_EXT	1579
6.410.2.2 VEC_TYPE	1579
6.410.3 Constructor & Destructor Documentation	1579
6.410.3.1 Minus_Vector_Primitive()	1579
6.410.4 Member Function Documentation	1579
6.410.4.1 operator[]()	1580
6.410.4.2 size()	1580

---

6.410.5 Member Data Documentation	1580
6.410.5.1 lhs	1580
6.410.5.2 rhs	1580
6.411 Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T > Class Template Reference	1581
6.411.1 Detailed Description	1582
6.411.2 Member Typedef Documentation	1583
6.411.2.1 STORAGE	1583
6.411.2.2 VEC_EXT	1583
6.411.2.3 VEC_TYPE	1583
6.411.3 Constructor & Destructor Documentation	1583
6.411.3.1 Minus_Vector_Primitive()	1583
6.411.4 Member Function Documentation	1584
6.411.4.1 operator[]()	1584
6.411.4.2 size()	1584
6.411.5 Member Data Documentation	1584
6.411.5.1 lhs	1584
6.411.5.2 rhs	1584
6.411.5.3 rhs_na	1585
6.412 Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	1585
6.412.1 Detailed Description	1586
6.412.2 Member Typedef Documentation	1586
6.412.2.1 LHS_EXT	1586
6.412.2.2 LHS_TYPE	1586
6.412.2.3 RHS_EXT	1587
6.412.2.4 RHS_TYPE	1587
6.412.2.5 STORAGE	1587
6.412.3 Constructor & Destructor Documentation	1587
6.412.3.1 Minus_Vector_Vector()	1587
6.412.4 Member Function Documentation	1587
6.412.4.1 operator[]()	1588
6.412.4.2 size()	1588
6.412.5 Member Data Documentation	1588
6.412.5.1 lhs	1588
6.412.5.2 rhs	1589
6.413 Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > Class Template Reference	1589
6.413.1 Detailed Description	1590
6.413.2 Member Typedef Documentation	1590
6.413.2.1 LHS_EXT	1590



---

6.413.2.2 LHS_TYPE . . . . .	1590
6.413.2.3 RHS_EXT . . . . .	1591
6.413.2.4 RHS_TYPE . . . . .	1591
6.413.3 Constructor & Destructor Documentation . . . . .	1591
6.413.3.1 Minus_Vector_Vector() . . . . .	1591
6.413.4 Member Function Documentation . . . . .	1591
6.413.4.1 operator[]() . . . . .	1591
6.413.4.2 size() . . . . .	1592
6.413.5 Member Data Documentation . . . . .	1592
6.413.5.1 lhs . . . . .	1592
6.413.5.2 rhs . . . . .	1592
6.414 Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > Class Template Reference . . . . .	1593
6.414.1 Detailed Description . . . . .	1594
6.414.2 Member Typedef Documentation . . . . .	1594
6.414.2.1 LHS_EXT . . . . .	1594
6.414.2.2 LHS_TYPE . . . . .	1594
6.414.2.3 RHS_EXT . . . . .	1594
6.414.2.4 RHS_TYPE . . . . .	1595
6.414.3 Constructor & Destructor Documentation . . . . .	1595
6.414.3.1 Minus_Vector_Vector() . . . . .	1595
6.414.4 Member Function Documentation . . . . .	1595
6.414.4.1 operator[]() . . . . .	1595
6.414.4.2 size() . . . . .	1595
6.414.5 Member Data Documentation . . . . .	1596
6.414.5.1 lhs . . . . .	1596
6.414.5.2 rhs . . . . .	1596
6.415 Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > Class Template Reference . . . . .	1596
6.415.1 Detailed Description . . . . .	1597
6.415.2 Member Typedef Documentation . . . . .	1597
6.415.2.1 LHS_EXT . . . . .	1597
6.415.2.2 LHS_TYPE . . . . .	1598
6.415.2.3 RHS_EXT . . . . .	1598
6.415.2.4 RHS_TYPE . . . . .	1598
6.415.3 Constructor & Destructor Documentation . . . . .	1598
6.415.3.1 Minus_Vector_Vector() . . . . .	1598
6.415.4 Member Function Documentation . . . . .	1598
6.415.4.1 operator[]() . . . . .	1599
6.415.4.2 size() . . . . .	1599

6.415.5 Member Data Documentation	1599
6.415.5.1 lhs	1599
6.415.5.2 rhs	1599
6.416 Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	1600
6.416.1 Detailed Description	1601
6.416.2 Member Typedef Documentation	1601
6.416.2.1 LHS_EXT	1601
6.416.2.2 LHS_TYPE	1601
6.416.2.3 RHS_EXT	1601
6.416.2.4 RHS_TYPE	1602
6.416.3 Constructor & Destructor Documentation	1602
6.416.3.1 Minus_Vector_Vector()	1602
6.416.4 Member Function Documentation	1602
6.416.4.1 operator[]()	1602
6.416.4.2 size()	1602
6.416.5 Member Data Documentation	1603
6.416.5.1 lhs	1603
6.416.5.2 rhs	1603
6.417 Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > Class Template Reference	1603
6.417.1 Detailed Description	1604
6.417.2 Member Typedef Documentation	1604
6.417.2.1 LHS_EXT	1604
6.417.2.2 LHS_TYPE	1605
6.417.2.3 RHS_EXT	1605
6.417.2.4 RHS_TYPE	1605
6.417.2.5 STORAGE	1605
6.417.3 Constructor & Destructor Documentation	1605
6.417.3.1 Minus_Vector_Vector()	1606
6.417.4 Member Function Documentation	1606
6.417.4.1 operator[]()	1606
6.417.4.2 size()	1606
6.417.5 Member Data Documentation	1606
6.417.5.1 lhs	1606
6.417.5.2 rhs	1607
6.418 Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > Class Template Reference	1607
6.418.1 Detailed Description	1608
6.418.2 Member Typedef Documentation	1608
6.418.2.1 LHS_EXT	1608

---

6.418.2.2 LHS_TYPE . . . . .	1608
6.418.2.3 RHS_EXT . . . . .	1609
6.418.2.4 RHS_TYPE . . . . .	1609
6.418.2.5 STORAGE . . . . .	1609
6.418.3 Constructor & Destructor Documentation . . . . .	1609
6.418.3.1 Minus_Vector_Vector() . . . . .	1609
6.418.4 Member Function Documentation . . . . .	1609
6.418.4.1 operator[]() . . . . .	1610
6.418.4.2 size() . . . . .	1610
6.418.5 Member Data Documentation . . . . .	1610
6.418.5.1 lhs . . . . .	1610
6.418.5.2 rhs . . . . .	1610
6.419 Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > Class Template Reference . . . . .	1611
6.419.1 Detailed Description . . . . .	1612
6.419.2 Member Typedef Documentation . . . . .	1612
6.419.2.1 LHS_EXT . . . . .	1612
6.419.2.2 LHS_TYPE . . . . .	1612
6.419.2.3 RHS_EXT . . . . .	1612
6.419.2.4 RHS_TYPE . . . . .	1613
6.419.2.5 STORAGE . . . . .	1613
6.419.3 Constructor & Destructor Documentation . . . . .	1613
6.419.3.1 Minus_Vector_Vector() . . . . .	1613
6.419.4 Member Function Documentation . . . . .	1613
6.419.4.1 operator[]() . . . . .	1613
6.419.4.2 size() . . . . .	1614
6.419.5 Member Data Documentation . . . . .	1614
6.419.5.1 lhs . . . . .	1614
6.419.5.2 rhs . . . . .	1614
6.420 Rcpp::Module Class Reference . . . . .	1614
6.420.1 Detailed Description . . . . .	1615
6.420.2 Member Typedef Documentation . . . . .	1615
6.420.2.1 CLASS_ITERATOR . . . . .	1615
6.420.2.2 CLASS_MAP . . . . .	1616
6.420.2.3 CLASS_PAIR . . . . .	1616
6.420.2.4 FUNCTION_PAIR . . . . .	1616
6.420.2.5 MAP . . . . .	1616
6.420.3 Constructor & Destructor Documentation . . . . .	1616
6.420.3.1 Module() [1/2] . . . . .	1616

---

6.420.3.2 Module() [2/2]	1617
6.420.4 Member Function Documentation	1617
6.420.4.1 Add()	1617
6.420.4.2 add_enum()	1618
6.420.4.3 AddClass()	1618
6.420.4.4 class_names()	1618
6.420.4.5 classes_info()	1619
6.420.4.6 complete()	1619
6.420.4.7 functions_arity()	1619
6.420.4.8 functions_names()	1620
6.420.4.9 get_class()	1620
6.420.4.10 get_class_pointer()	1620
6.420.4.11 get_function()	1620
6.420.4.12 get_function_ptr()	1621
6.420.4.13 has_class()	1622
6.420.4.14 has_function()	1622
6.420.4.15 invoke()	1622
6.420.5 Member Data Documentation	1623
6.420.5.1 classes	1623
6.420.5.2 functions	1624
6.420.5.3 name	1624
6.420.5.4 prefix	1624
6.421 Rcpp::traits::module_wrap_traits< T > Struct Template Reference	1624
6.421.1 Detailed Description	1625
6.421.2 Member Typedef Documentation	1625
6.421.2.1 category	1625
6.422 Rcpp::traits::module_wrap_traits< T * > Struct Template Reference	1625
6.422.1 Detailed Description	1625
6.422.2 Member Typedef Documentation	1625
6.422.2.1 category	1626
6.423 Rcpp::traits::module_wrap_traits< void > Struct Reference	1626
6.423.1 Detailed Description	1626
6.423.2 Member Typedef Documentation	1626
6.423.2.1 category	1626
6.424 Rcpp::Na_Proxy Class Reference	1626
6.424.1 Detailed Description	1627
6.424.2 Friends And Related Function Documentation	1627
6.424.2.1 operator== [1/18]	1627
6.424.2.2 operator== [2/18]	1627

---

6.424.2.3 operator== [3/18]	1628
6.424.2.4 operator== [4/18]	1628
6.424.2.5 operator== [5/18]	1628
6.424.2.6 operator== [6/18]	1628
6.424.2.7 operator== [7/18]	1628
6.424.2.8 operator== [8/18]	1629
6.424.2.9 operator== [9/18]	1629
6.424.2.10 operator== [10/18]	1629
6.424.2.11 operator== [11/18]	1629
6.424.2.12 operator== [12/18]	1629
6.424.2.13 operator== [13/18]	1630
6.424.2.14 operator== [14/18]	1630
6.424.2.15 operator== [15/18]	1630
6.424.2.16 operator== [16/18]	1630
6.424.2.17 operator== [17/18]	1630
6.424.2.18 operator== [18/18]	1631
6.425 Rcpp::traits::named_object< T > Class Template Reference	1631
6.425.1 Detailed Description	1631
6.425.2 Constructor & Destructor Documentation	1631
6.425.2.1 named_object()	1631
6.425.3 Member Data Documentation	1632
6.425.3.1 name	1632
6.425.3.2 object	1632
6.426 Rcpp::traits::named_object< SEXP > Class Reference	1632
6.426.1 Detailed Description	1633
6.426.2 Constructor & Destructor Documentation	1633
6.426.2.1 named_object() [1/2]	1633
6.426.2.2 named_object() [2/2]	1633
6.426.2.3 ~named_object()	1634
6.426.3 Member Data Documentation	1634
6.426.3.1 name	1634
6.426.3.2 object	1634
6.426.3.3 token	1634
6.427 Rcpp::internal::NamedPlaceholder Class Reference	1635
6.427.1 Detailed Description	1635
6.427.2 Constructor & Destructor Documentation	1635
6.427.2.1 NamedPlaceholder()	1635
6.427.2.2 ~NamedPlaceholder()	1635
6.427.3 Member Function Documentation	1635

---

6.427.3.1 operator SEXP() . . . . .	1636
6.427.3.2 operator>() . . . . .	1636
6.427.3.3 operator[]() . . . . .	1636
6.428 Rcpp::NamesProxyPolicy< CLASS >::NamesProxy Class Reference . . . . .	1636
6.428.1 Detailed Description . . . . .	1637
6.428.2 Constructor & Destructor Documentation . . . . .	1638
6.428.2.1 NamesProxy() . . . . .	1638
6.428.3 Member Function Documentation . . . . .	1638
6.428.3.1 get() . . . . .	1638
6.428.3.2 operator T() . . . . .	1638
6.428.3.3 operator=() [1/3] . . . . .	1639
6.428.3.4 operator=() [2/3] . . . . .	1639
6.428.3.5 operator=() [3/3] . . . . .	1640
6.428.3.6 set() . . . . .	1640
6.428.4 Member Data Documentation . . . . .	1640
6.428.4.1 parent . . . . .	1640
6.429 Rcpp::NamesProxyPolicy< CLASS > Class Template Reference . . . . .	1641
6.429.1 Detailed Description . . . . .	1641
6.429.2 Member Function Documentation . . . . .	1642
6.429.2.1 names() [1/2] . . . . .	1642
6.429.2.2 names() [2/2] . . . . .	1642
6.430 Rcpp::stats::NBinomGenerator Class Reference . . . . .	1642
6.430.1 Detailed Description . . . . .	1643
6.430.2 Constructor & Destructor Documentation . . . . .	1643
6.430.2.1 NBinomGenerator() . . . . .	1643
6.430.3 Member Function Documentation . . . . .	1644
6.430.3.1 operator>() . . . . .	1644
6.430.4 Member Data Documentation . . . . .	1644
6.430.4.1 lambda . . . . .	1644
6.430.4.2 siz . . . . .	1644
6.431 Rcpp::stats::NBinomGenerator_Mu Class Reference . . . . .	1645
6.431.1 Detailed Description . . . . .	1646
6.431.2 Constructor & Destructor Documentation . . . . .	1646
6.431.2.1 NBinomGenerator_Mu() . . . . .	1646
6.431.3 Member Function Documentation . . . . .	1646
6.431.3.1 operator>() . . . . .	1646
6.431.4 Member Data Documentation . . . . .	1646
6.431.4.1 lambda . . . . .	1646
6.431.4.2 siz . . . . .	1647

---

6.432 Rcpp::stats::NChisqGenerator Class Reference	1647
6.432.1 Detailed Description	1648
6.432.2 Constructor & Destructor Documentation	1648
6.432.2.1 NChisqGenerator()	1648
6.432.3 Member Function Documentation	1648
6.432.3.1 operator>()	1648
6.432.4 Member Data Documentation	1649
6.432.4.1 df	1649
6.432.4.2 df_2	1649
6.432.4.3 lambda_2	1649
6.433 Rcpp::traits::needs_protection< T > Struct Template Reference	1650
6.433.1 Detailed Description	1650
6.434 Rcpp::traits::needs_protection< SEXP > Struct Reference	1651
6.434.1 Detailed Description	1651
6.435 Rcpp::sugar::negate< NA > Struct Template Reference	1652
6.435.1 Detailed Description	1652
6.435.2 Member Function Documentation	1652
6.435.2.1 apply()	1652
6.436 Rcpp::sugar::negate< false > Struct Reference	1653
6.436.1 Detailed Description	1653
6.436.2 Member Function Documentation	1653
6.436.2.1 apply()	1653
6.437 Rcpp::sugar::Negate_SingleLogicalResult< NA, T > Class Template Reference	1653
6.437.1 Detailed Description	1655
6.437.2 Member Typedef Documentation	1655
6.437.2.1 BASE	1655
6.437.2.2 TYPE	1655
6.437.3 Constructor & Destructor Documentation	1655
6.437.3.1 Negate_SingleLogicalResult()	1655
6.437.4 Member Function Documentation	1656
6.437.4.1 apply()	1656
6.437.5 Member Data Documentation	1656
6.437.5.1 orig	1656
6.438 Rcpp::newDatetimeVector Class Reference	1657
6.438.1 Detailed Description	1658
6.438.2 Constructor & Destructor Documentation	1658
6.438.2.1 newDatetimeVector() [1/3]	1658
6.438.2.2 newDatetimeVector() [2/3]	1659
6.438.2.3 newDatetimeVector() [3/3]	1659

---

6.438.3 Member Function Documentation	1659
6.438.3.1 getDatetimes()	1660
6.438.3.2 operator=()	1660
6.438.3.3 setClass()	1661
6.438.4 Friends And Related Function Documentation	1661
6.438.4.1 operator<<	1661
6.439 Rcpp::newDateVector Class Reference	1661
6.439.1 Detailed Description	1662
6.439.2 Constructor & Destructor Documentation	1662
6.439.2.1 newDateVector() [1/3]	1663
6.439.2.2 newDateVector() [2/3]	1663
6.439.2.3 newDateVector() [3/3]	1664
6.439.3 Member Function Documentation	1664
6.439.3.1 getDates()	1664
6.439.3.2 operator=()	1665
6.439.3.3 setClass()	1665
6.439.4 Friends And Related Function Documentation	1665
6.439.4.1 operator<<	1665
6.440 Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::no Struct Reference	1666
6.440.1 Detailed Description	1666
6.440.2 Member Data Documentation	1666
6.440.2.1 array	1666
6.441 Rcpp::no_init_matrix Class Reference	1666
6.441.1 Detailed Description	1667
6.441.2 Constructor & Destructor Documentation	1667
6.441.2.1 no_init_matrix()	1667
6.441.3 Member Function Documentation	1667
6.441.3.1 ncol()	1667
6.441.3.2 nrow()	1667
6.441.3.3 operator Matrix< RTYPE, StoragePolicy >()	1668
6.441.4 Member Data Documentation	1668
6.441.4.1 nc	1668
6.441.4.2 nr	1668
6.442 Rcpp::no_init_vector Class Reference	1668
6.442.1 Detailed Description	1669
6.442.2 Constructor & Destructor Documentation	1669
6.442.2.1 no_init_vector()	1669
6.442.3 Member Function Documentation	1669
6.442.3.1 get()	1669



---

6.442.3.2 operator Vector< RTYPE, StoragePolicy >()	1669
6.442.4 Member Data Documentation	1670
6.442.4.1 size	1670
6.443 Rcpp::no_such_env Class Reference	1670
6.443.1 Detailed Description	1671
6.443.2 Constructor & Destructor Documentation	1671
6.443.2.1 no_such_env() [1/2]	1671
6.443.2.2 no_such_env() [2/2]	1671
6.443.2.3 ~no_such_env()	1671
6.443.3 Member Function Documentation	1672
6.443.3.1 what()	1672
6.443.4 Member Data Documentation	1672
6.443.4.1 message	1672
6.444 Rcpp::sugar::Nona< RTYPE, NA, VECTOR > Class Template Reference	1672
6.444.1 Detailed Description	1674
6.444.2 Member Typedef Documentation	1674
6.444.2.1 STORAGE	1674
6.444.2.2 SUGAR_TYPE	1674
6.444.3 Constructor & Destructor Documentation	1674
6.444.3.1 Nona()	1675
6.444.4 Member Function Documentation	1675
6.444.4.1 operator[]()	1675
6.444.4.2 size()	1675
6.444.5 Member Data Documentation	1675
6.444.5.1 data	1675
6.445 Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > > Class Template Reference	1676
6.445.1 Detailed Description	1677
6.445.2 Member Typedef Documentation	1678
6.445.2.1 iterator	1678
6.445.2.2 STORAGE	1678
6.445.2.3 SUGAR_TYPE	1678
6.445.3 Constructor & Destructor Documentation	1678
6.445.3.1 Nona()	1678
6.445.4 Member Function Documentation	1679
6.445.4.1 operator[]()	1679
6.445.4.2 size()	1679
6.445.5 Member Data Documentation	1679
6.445.5.1 data	1679
6.445.5.2 n	1679

---

6.446 Rcpp::sugar::NonaPrimitive< T > Class Template Reference . . . . .	1680
6.446.1 Detailed Description . . . . .	1680
6.446.2 Constructor & Destructor Documentation . . . . .	1680
6.446.2.1 NonaPrimitive() . . . . .	1680
6.446.3 Member Function Documentation . . . . .	1680
6.446.3.1 operator T() . . . . .	1680
6.446.4 Member Data Documentation . . . . .	1681
6.446.4.1 x . . . . .	1681
6.447 Rcpp::NoProtectStorage< CLASS > Class Template Reference . . . . .	1681
6.447.1 Detailed Description . . . . .	1681
6.447.2 Constructor & Destructor Documentation . . . . .	1682
6.447.2.1 NoProtectStorage() . . . . .	1682
6.447.2.2 ~NoProtectStorage() . . . . .	1682
6.447.3 Member Function Documentation . . . . .	1682
6.447.3.1 copy__() . . . . .	1682
6.447.3.2 get__() . . . . .	1683
6.447.3.3 inherits() . . . . .	1683
6.447.3.4 invalidate__() . . . . .	1683
6.447.3.5 operator SEXP() . . . . .	1683
6.447.3.6 set__() . . . . .	1684
6.447.4 Member Data Documentation . . . . .	1684
6.447.4.1 data . . . . .	1684
6.448 Rcpp::traits::normal_wrap_tag Struct Reference . . . . .	1684
6.448.1 Detailed Description . . . . .	1684
6.449 Rcpp::stats::NormGenerator Class Reference . . . . .	1685
6.449.1 Detailed Description . . . . .	1686
6.449.2 Constructor & Destructor Documentation . . . . .	1686
6.449.2.1 NormGenerator() . . . . .	1686
6.449.3 Member Function Documentation . . . . .	1686
6.449.3.1 operator>() . . . . .	1686
6.449.4 Member Data Documentation . . . . .	1687
6.449.4.1 mean . . . . .	1687
6.449.4.2 sd . . . . .	1687
6.450 Rcpp::stats::NormGenerator__mean0 Class Reference . . . . .	1687
6.450.1 Detailed Description . . . . .	1688
6.450.2 Constructor & Destructor Documentation . . . . .	1688
6.450.2.1 NormGenerator__mean0() . . . . .	1688
6.450.3 Member Function Documentation . . . . .	1689
6.450.3.1 operator>() . . . . .	1689

---

6.450.4 Member Data Documentation	1689
6.450.4.1 sd	1689
6.451 Rcpp::stats::NormGenerator__mean0__sd1 Class Reference	1690
6.451.1 Detailed Description	1690
6.451.2 Constructor & Destructor Documentation	1691
6.451.2.1 NormGenerator__mean0__sd1()	1691
6.451.3 Member Function Documentation	1691
6.451.3.1 operator>()	1691
6.452 Rcpp::stats::NormGenerator__sd1 Class Reference	1692
6.452.1 Detailed Description	1693
6.452.2 Constructor & Destructor Documentation	1693
6.452.2.1 NormGenerator__sd1()	1693
6.452.3 Member Function Documentation	1693
6.452.3.1 operator>()	1693
6.452.4 Member Data Documentation	1694
6.452.4.1 mean	1694
6.453 Rcpp::sugar::not_< RTYPE, NA > Class Template Reference	1694
6.453.1 Detailed Description	1694
6.453.2 Member Typedef Documentation	1694
6.453.2.1 STORAGE	1695
6.453.3 Member Function Documentation	1695
6.453.3.1 apply()	1695
6.454 Rcpp::sugar::not_< CPLXSP, false > Class Reference	1695
6.454.1 Detailed Description	1695
6.454.2 Member Function Documentation	1695
6.454.2.1 apply()	1696
6.455 Rcpp::sugar::not_< CPLXSP, NA > Class Template Reference	1696
6.455.1 Detailed Description	1696
6.455.2 Member Function Documentation	1696
6.455.2.1 apply()	1697
6.456 Rcpp::sugar::not_< REALSP, false > Class Reference	1697
6.456.1 Detailed Description	1697
6.456.2 Member Function Documentation	1697
6.456.2.1 apply()	1698
6.457 Rcpp::sugar::not_< REALSP, NA > Class Template Reference	1698
6.457.1 Detailed Description	1698
6.457.2 Member Function Documentation	1698
6.457.2.1 apply()	1699
6.458 Rcpp::sugar::not_< RTYPE, false > Class Template Reference	1699

---

6.458.1 Detailed Description	1699
6.458.2 Member Typedef Documentation	1700
6.458.2.1 STORAGE	1700
6.458.3 Member Function Documentation	1700
6.458.3.1 apply()	1700
6.459 Rcpp::sugar::Not_Vector< RTYPE, NA, T > Class Template Reference	1700
6.459.1 Detailed Description	1702
6.459.2 Member Typedef Documentation	1702
6.459.2.1 OPERATOR	1702
6.459.2.2 STORAGE	1702
6.459.2.3 VEC_TYPE	1702
6.459.3 Constructor & Destructor Documentation	1702
6.459.3.1 Not_Vector()	1703
6.459.4 Member Function Documentation	1703
6.459.4.1 operator[]()	1703
6.459.4.2 size()	1704
6.459.5 Member Data Documentation	1704
6.459.5.1 lhs	1704
6.459.5.2 op	1704
6.460 Rcpp::Nullable< T > Class Template Reference	1705
6.460.1 Detailed Description	1705
6.460.2 Constructor & Destructor Documentation	1705
6.460.2.1 Nullable() [1/3]	1706
6.460.2.2 Nullable() [2/3]	1706
6.460.2.3 Nullable() [3/3]	1706
6.460.3 Member Function Documentation	1706
6.460.3.1 as()	1707
6.460.3.2 checkIfSet()	1707
6.460.3.3 clone()	1707
6.460.3.4 get()	1708
6.460.3.5 isNotNull()	1708
6.460.3.6 isNull()	1709
6.460.3.7 isSet()	1710
6.460.3.8 isUsable()	1710
6.460.3.9 operator SEXP()	1710
6.460.3.10 operator=()	1711
6.460.4 Member Data Documentation	1711
6.460.4.1 m_set	1711
6.460.4.2 m_sexp	1712

---

6.461 Rcpp::traits::num2type< N > Struct Template Reference	1712
6.461.1 Detailed Description	1712
6.462 Rcpp::object< T > Class Template Reference	1712
6.462.1 Detailed Description	1713
6.462.2 Member Typedef Documentation	1713
6.462.2.1 object_type	1713
6.462.3 Constructor & Destructor Documentation	1713
6.462.3.1 object()	1713
6.462.4 Member Function Documentation	1713
6.462.4.1 operator T*()	1714
6.462.4.2 operator&()	1714
6.462.4.3 operator->()	1714
6.462.5 Member Data Documentation	1714
6.462.5.1 ptr	1714
6.463 Rcpp::oldDatetimeVector Class Reference	1715
6.463.1 Detailed Description	1715
6.463.2 Constructor & Destructor Documentation	1716
6.463.2.1 oldDatetimeVector() [1/2]	1716
6.463.2.2 oldDatetimeVector() [2/2]	1716
6.463.3 Member Function Documentation	1716
6.463.3.1 getDatetimes()	1716
6.464 Rcpp::oldDateVector Class Reference	1717
6.464.1 Detailed Description	1717
6.464.2 Constructor & Destructor Documentation	1718
6.464.2.1 oldDateVector() [1/2]	1718
6.464.2.2 oldDateVector() [2/2]	1718
6.464.3 Member Function Documentation	1718
6.464.3.1 getDates()	1718
6.465 Rcpp::traits::one_type< T > Class Template Reference	1718
6.465.1 Detailed Description	1719
6.465.2 Member Function Documentation	1719
6.465.2.1 op() [1/2]	1719
6.465.2.2 op() [2/2]	1719
6.465.2.3 operator T()	1720
6.466 Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	1720
6.466.1 Detailed Description	1721
6.466.2 Member Typedef Documentation	1721
6.466.2.1 LHS_TYPE	1721

---

6.466.2.2 RHS_TYPE . . . . .	1721
6.466.3 Constructor & Destructor Documentation . . . . .	1722
6.466.3.1 Or_LogicalExpression_LogicalExpression() . . . . .	1722
6.466.4 Member Function Documentation . . . . .	1722
6.466.4.1 operator[]() . . . . .	1722
6.466.4.2 size() . . . . .	1722
6.466.5 Member Data Documentation . . . . .	1723
6.466.5.1 lhs . . . . .	1723
6.466.5.2 rhs . . . . .	1723
6.467 Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > Class Template Reference . . . . .	1724
6.467.1 Detailed Description . . . . .	1725
6.467.2 Member Typedef Documentation . . . . .	1725
6.467.2.1 LHS_TYPE . . . . .	1725
6.467.2.2 RHS_TYPE . . . . .	1725
6.467.3 Constructor & Destructor Documentation . . . . .	1725
6.467.3.1 Or_LogicalExpression_LogicalExpression() . . . . .	1725
6.467.4 Member Function Documentation . . . . .	1726
6.467.4.1 operator[]() . . . . .	1726
6.467.4.2 size() . . . . .	1726
6.467.5 Member Data Documentation . . . . .	1726
6.467.5.1 lhs . . . . .	1727
6.467.5.2 rhs . . . . .	1727
6.468 Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > Class Template Reference . . . . .	1727
6.468.1 Detailed Description . . . . .	1728
6.468.2 Member Typedef Documentation . . . . .	1728
6.468.2.1 LHS_TYPE . . . . .	1728
6.468.2.2 RHS_TYPE . . . . .	1728
6.468.3 Constructor & Destructor Documentation . . . . .	1729
6.468.3.1 Or_LogicalExpression_LogicalExpression() . . . . .	1729
6.468.4 Member Function Documentation . . . . .	1729
6.468.4.1 operator[]() . . . . .	1729
6.468.4.2 size() . . . . .	1729
6.468.5 Member Data Documentation . . . . .	1730
6.468.5.1 lhs . . . . .	1730
6.468.5.2 rhs . . . . .	1730
6.469 Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > Class Template Reference . . . . .	1730
6.469.1 Detailed Description . . . . .	1731

---

6.469.2 Member Typedef Documentation	1731
6.469.2.1 LHS_TYPE	1731
6.469.2.2 RHS_TYPE	1731
6.469.3 Constructor & Destructor Documentation	1732
6.469.3.1 Or_LogicalExpression_LogicalExpression()	1732
6.469.4 Member Function Documentation	1732
6.469.4.1 operator[]()	1732
6.469.4.2 size()	1732
6.469.5 Member Data Documentation	1733
6.469.5.1 lhs	1733
6.469.5.2 rhs	1733
6.470 Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T > Class Template Reference	1733
6.470.1 Detailed Description	1734
6.470.2 Member Typedef Documentation	1734
6.470.2.1 BASE	1735
6.470.2.2 LHS_TYPE	1735
6.470.3 Constructor & Destructor Documentation	1735
6.470.3.1 Or_SingleLogicalResult_bool()	1735
6.470.4 Member Function Documentation	1735
6.470.4.1 apply()	1736
6.470.5 Member Data Documentation	1736
6.470.5.1 lhs	1736
6.470.5.2 rhs	1736
6.471 Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	1737
6.471.1 Detailed Description	1738
6.471.2 Member Typedef Documentation	1738
6.471.2.1 BASE	1738
6.471.2.2 LHS_TYPE	1738
6.471.2.3 RHS_TYPE	1738
6.471.3 Constructor & Destructor Documentation	1739
6.471.3.1 Or_SingleLogicalResult_SingleLogicalResult()	1739
6.471.4 Member Function Documentation	1739
6.471.4.1 apply()	1739
6.471.5 Member Data Documentation	1740
6.471.5.1 lhs	1740
6.471.5.2 rhs	1740
6.472 Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T > Class Template Reference	1741
6.472.1 Detailed Description	1742

6.472.2 Member Typedef Documentation	1742
6.472.2.1 BASE	1742
6.472.2.2 LHS_TYPE	1742
6.472.2.3 RHS_TYPE	1742
6.472.3 Constructor & Destructor Documentation	1743
6.472.3.1 Or_SingleLogicalResult_SingleLogicalResult()	1743
6.472.4 Member Function Documentation	1743
6.472.4.1 apply()	1743
6.472.5 Member Data Documentation	1743
6.472.5.1 lhs	1744
6.472.5.2 rhs	1744
6.473 Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > Class Template Reference	1744
6.473.1 Detailed Description	1745
6.473.2 Member Typedef Documentation	1745
6.473.2.1 BASE	1745
6.473.2.2 LHS_TYPE	1746
6.473.2.3 RHS_TYPE	1746
6.473.3 Constructor & Destructor Documentation	1746
6.473.3.1 Or_SingleLogicalResult_SingleLogicalResult()	1746
6.473.4 Member Function Documentation	1746
6.473.4.1 apply()	1747
6.473.5 Member Data Documentation	1747
6.473.5.1 lhs	1747
6.473.5.2 rhs	1747
6.474 Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > Class Template Reference	1748
6.474.1 Detailed Description	1749
6.474.2 Member Typedef Documentation	1749
6.474.2.1 BASE	1749
6.474.2.2 LHS_TYPE	1749
6.474.2.3 RHS_TYPE	1749
6.474.3 Constructor & Destructor Documentation	1750
6.474.3.1 Or_SingleLogicalResult_SingleLogicalResult()	1750
6.474.4 Member Function Documentation	1750
6.474.4.1 apply()	1750
6.474.5 Member Data Documentation	1750
6.474.5.1 lhs	1751
6.474.5.2 rhs	1751
6.475 Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function > Class Template Reference	1751



---

6.475.1 Detailed Description	1752
6.475.2 Member Typedef Documentation	1752
6.475.2.1 converter_type	1753
6.475.2.2 LHS_LAZY	1753
6.475.2.3 LHS_TYPE	1753
6.475.2.4 result_type	1753
6.475.2.5 RHS_LAZY	1754
6.475.2.6 RHS_TYPE	1754
6.475.2.7 STORAGE	1754
6.475.3 Constructor & Destructor Documentation	1754
6.475.3.1 Outer()	1754
6.475.4 Member Function Documentation	1755
6.475.4.1 ncol()	1755
6.475.4.2 nrow()	1755
6.475.4.3 operator>()	1755
6.475.4.4 size()	1756
6.475.5 Member Data Documentation	1756
6.475.5.1 fun	1756
6.475.5.2 lhs	1756
6.475.5.3 nc	1756
6.475.5.4 nr	1757
6.475.5.5 RESULT_R_TYPE	1757
6.475.5.6 rhs	1757
6.476 Rcpp::stats::P0< RTYPE, NA, T > Class Template Reference	1758
6.476.1 Detailed Description	1759
6.476.2 Member Typedef Documentation	1759
6.476.2.1 FunPtr	1759
6.476.2.2 VEC_TYPE	1759
6.476.3 Constructor & Destructor Documentation	1759
6.476.3.1 P0()	1760
6.476.4 Member Function Documentation	1760
6.476.4.1 operator[]()	1760
6.476.4.2 size()	1760
6.476.5 Member Data Documentation	1761
6.476.5.1 log	1761
6.476.5.2 lower	1761
6.476.5.3 ptr	1761
6.476.5.4 vec	1761
6.477 Rcpp::stats::P1< RTYPE, NA, T > Class Template Reference	1762

---

6.477.1 Detailed Description	1763
6.477.2 Member Typedef Documentation	1763
6.477.2.1 FunPtr	1763
6.477.2.2 VEC_TYPE	1763
6.477.3 Constructor & Destructor Documentation	1763
6.477.3.1 P1()	1764
6.477.4 Member Function Documentation	1764
6.477.4.1 operator[]()	1764
6.477.4.2 size()	1764
6.477.5 Member Data Documentation	1765
6.477.5.1 log	1765
6.477.5.2 lower	1765
6.477.5.3 p0	1765
6.477.5.4 ptr	1765
6.477.5.5 vec	1766
6.478 Rcpp::stats::P2< RTYPE, NA, T > Class Template Reference	1766
6.478.1 Detailed Description	1767
6.478.2 Member Typedef Documentation	1767
6.478.2.1 FunPtr	1767
6.478.2.2 VEC_TYPE	1768
6.478.3 Constructor & Destructor Documentation	1768
6.478.3.1 P2()	1768
6.478.4 Member Function Documentation	1768
6.478.4.1 operator[]()	1768
6.478.4.2 size()	1769
6.478.5 Member Data Documentation	1769
6.478.5.1 log	1769
6.478.5.2 lower	1769
6.478.5.3 p0	1770
6.478.5.4 p1	1770
6.478.5.5 ptr	1770
6.478.5.6 vec	1770
6.479 Rcpp::stats::P3< RTYPE, NA, T > Class Template Reference	1771
6.479.1 Detailed Description	1772
6.479.2 Member Typedef Documentation	1772
6.479.2.1 FunPtr	1772
6.479.2.2 VEC_TYPE	1772
6.479.3 Constructor & Destructor Documentation	1772
6.479.3.1 P3()	1773

---

6.479.4 Member Function Documentation	1773
6.479.4.1 operator[]()	1773
6.479.4.2 size()	1773
6.479.5 Member Data Documentation	1774
6.479.5.1 log	1774
6.479.5.2 lower	1774
6.479.5.3 p0	1774
6.479.5.4 p1	1774
6.479.5.5 p2	1775
6.479.5.6 ptr	1775
6.479.5.7 vec	1775
6.480 Rcpp::attributes::Param Class Reference	1775
6.480.1 Detailed Description	1776
6.480.2 Constructor & Destructor Documentation	1776
6.480.2.1 Param() [1/2]	1776
6.480.2.2 Param() [2/2]	1776
6.480.3 Member Function Documentation	1777
6.480.3.1 empty()	1777
6.480.3.2 name()	1777
6.480.3.3 operator"!="()	1777
6.480.3.4 operator==(())	1778
6.480.3.5 value()	1778
6.480.4 Member Data Documentation	1778
6.480.4.1 name_	1778
6.480.4.2 value_	1778
6.481 Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T > Class Template Reference	1779
6.481.1 Detailed Description	1780
6.481.2 Member Typedef Documentation	1781
6.481.2.1 EXT	1781
6.481.2.2 STORAGE	1781
6.481.2.3 VEC_TYPE	1781
6.481.3 Constructor & Destructor Documentation	1781
6.481.3.1 Plus_Vector_Primitive()	1781
6.481.4 Member Function Documentation	1782
6.481.4.1 operator[]()	1782
6.481.4.2 size()	1782
6.481.5 Member Data Documentation	1782
6.481.5.1 lhs	1782
6.481.5.2 rhs	1783

6.481.5.3 rhs_na	1783
6.482 Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T > Class Template Reference	1783
6.482.1 Detailed Description	1784
6.482.2 Member Typedef Documentation	1785
6.482.2.1 EXT	1785
6.482.2.2 VEC_TYPE	1785
6.482.3 Constructor & Destructor Documentation	1785
6.482.3.1 Plus_Vector_Primitive()	1785
6.482.4 Member Function Documentation	1785
6.482.4.1 operator[]()	1786
6.482.4.2 size()	1786
6.482.5 Member Data Documentation	1786
6.482.5.1 lhs	1786
6.482.5.2 rhs	1786
6.483 Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T > Class Template Reference	1787
6.483.1 Detailed Description	1788
6.483.2 Member Typedef Documentation	1789
6.483.2.1 EXT	1789
6.483.2.2 VEC_TYPE	1789
6.483.3 Constructor & Destructor Documentation	1789
6.483.3.1 Plus_Vector_Primitive()	1789
6.483.4 Member Function Documentation	1789
6.483.4.1 operator[]()	1790
6.483.4.2 size()	1790
6.483.5 Member Data Documentation	1790
6.483.5.1 lhs	1790
6.483.5.2 rhs	1790
6.484 Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T > Class Template Reference	1791
6.484.1 Detailed Description	1792
6.484.2 Member Typedef Documentation	1793
6.484.2.1 EXT	1793
6.484.2.2 STORAGE	1793
6.484.2.3 VEC_TYPE	1793
6.484.3 Constructor & Destructor Documentation	1793
6.484.3.1 Plus_Vector_Primitive()	1793
6.484.4 Member Function Documentation	1794
6.484.4.1 operator[]()	1794
6.484.4.2 size()	1794
6.484.5 Member Data Documentation	1794

---

6.484.5.1 lhs	1794
6.484.5.2 rhs	1794
6.484.5.3 rhs_na	1795
6.485 Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T > Class Template Reference	1795
6.485.1 Detailed Description	1796
6.485.2 Member Typedef Documentation	1797
6.485.2.1 EXT	1797
6.485.2.2 STORAGE	1797
6.485.2.3 VEC_TYPE	1797
6.485.3 Constructor & Destructor Documentation	1797
6.485.3.1 Plus_Vector_Primitive_nona()	1797
6.485.4 Member Function Documentation	1798
6.485.4.1 operator[]()	1798
6.485.4.2 size()	1798
6.485.5 Member Data Documentation	1798
6.485.5.1 lhs	1798
6.485.5.2 rhs	1799
6.486 Rcpp::sugar::Plus_Vector_Primitive_nona< REALXP, false, T > Class Template Reference	1799
6.486.1 Detailed Description	1800
6.486.2 Member Typedef Documentation	1800
6.486.2.1 EXT	1800
6.486.2.2 VEC_TYPE	1800
6.486.3 Constructor & Destructor Documentation	1800
6.486.3.1 Plus_Vector_Primitive_nona()	1801
6.486.4 Member Function Documentation	1801
6.486.4.1 operator[]()	1801
6.486.4.2 size()	1801
6.486.5 Member Data Documentation	1801
6.486.5.1 lhs	1801
6.486.5.2 rhs	1802
6.487 Rcpp::sugar::Plus_Vector_Primitive_nona< REALXP, NA, T > Class Template Reference	1802
6.487.1 Detailed Description	1803
6.487.2 Member Typedef Documentation	1804
6.487.2.1 EXT	1804
6.487.2.2 VEC_TYPE	1804
6.487.3 Constructor & Destructor Documentation	1804
6.487.3.1 Plus_Vector_Primitive_nona()	1804
6.487.4 Member Function Documentation	1804
6.487.4.1 operator[]()	1805

---

6.487.4.2 size()	1805
6.487.5 Member Data Documentation	1805
6.487.5.1 lhs	1805
6.487.5.2 rhs	1805
6.488 Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T > Class Template Reference	1806
6.488.1 Detailed Description	1807
6.488.2 Member Typedef Documentation	1808
6.488.2.1 EXT	1808
6.488.2.2 STORAGE	1808
6.488.2.3 VEC_TYPE	1808
6.488.3 Constructor & Destructor Documentation	1808
6.488.3.1 Plus_Vector_Primitive_nona()	1808
6.488.4 Member Function Documentation	1809
6.488.4.1 operator[]()	1809
6.488.4.2 size()	1809
6.488.5 Member Data Documentation	1809
6.488.5.1 lhs	1809
6.488.5.2 rhs	1809
6.489 Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	1810
6.489.1 Detailed Description	1811
6.489.2 Member Typedef Documentation	1811
6.489.2.1 LHS_EXT	1811
6.489.2.2 LHS_TYPE	1811
6.489.2.3 RHS_EXT	1811
6.489.2.4 RHS_TYPE	1812
6.489.2.5 STORAGE	1812
6.489.3 Constructor & Destructor Documentation	1812
6.489.3.1 Plus_Vector_Vector()	1812
6.489.4 Member Function Documentation	1812
6.489.4.1 operator[]()	1812
6.489.4.2 size()	1813
6.489.5 Member Data Documentation	1813
6.489.5.1 lhs	1813
6.489.5.2 rhs	1813
6.490 Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > Class Template Reference	1814
6.490.1 Detailed Description	1815
6.490.2 Member Typedef Documentation	1815
6.490.2.1 LHS_EXT	1815

---

6.490.2.2 LHS_TYPE . . . . .	1815
6.490.2.3 RHS_EXT . . . . .	1815
6.490.2.4 RHS_TYPE . . . . .	1816
6.490.3 Constructor & Destructor Documentation . . . . .	1816
6.490.3.1 Plus_Vector_Vector() . . . . .	1816
6.490.4 Member Function Documentation . . . . .	1816
6.490.4.1 operator[]() . . . . .	1816
6.490.4.2 size() . . . . .	1816
6.490.5 Member Data Documentation . . . . .	1817
6.490.5.1 lhs . . . . .	1817
6.490.5.2 rhs . . . . .	1817
6.491 Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > Class Template Reference . . . . .	1817
6.491.1 Detailed Description . . . . .	1818
6.491.2 Member Typedef Documentation . . . . .	1818
6.491.2.1 LHS_EXT . . . . .	1818
6.491.2.2 LHS_TYPE . . . . .	1819
6.491.2.3 RHS_EXT . . . . .	1819
6.491.2.4 RHS_TYPE . . . . .	1819
6.491.3 Constructor & Destructor Documentation . . . . .	1819
6.491.3.1 Plus_Vector_Vector() . . . . .	1819
6.491.4 Member Function Documentation . . . . .	1819
6.491.4.1 operator[]() . . . . .	1820
6.491.4.2 size() . . . . .	1820
6.491.5 Member Data Documentation . . . . .	1820
6.491.5.1 lhs . . . . .	1820
6.491.5.2 rhs . . . . .	1820
6.492 Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > Class Template Reference . . . . .	1821
6.492.1 Detailed Description . . . . .	1822
6.492.2 Member Typedef Documentation . . . . .	1822
6.492.2.1 LHS_EXT . . . . .	1822
6.492.2.2 LHS_TYPE . . . . .	1822
6.492.2.3 RHS_EXT . . . . .	1822
6.492.2.4 RHS_TYPE . . . . .	1823
6.492.3 Constructor & Destructor Documentation . . . . .	1823
6.492.3.1 Plus_Vector_Vector() . . . . .	1823
6.492.4 Member Function Documentation . . . . .	1823
6.492.4.1 operator[]() . . . . .	1823
6.492.4.2 size() . . . . .	1823

---

6.492.5 Member Data Documentation	1824
6.492.5.1 lhs	1824
6.492.5.2 rhs	1824
6.493 Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	1824
6.493.1 Detailed Description	1825
6.493.2 Member Typedef Documentation	1825
6.493.2.1 LHS_EXT	1825
6.493.2.2 LHS_TYPE	1826
6.493.2.3 RHS_EXT	1826
6.493.2.4 RHS_TYPE	1826
6.493.3 Constructor & Destructor Documentation	1826
6.493.3.1 Plus_Vector_Vector()	1826
6.493.4 Member Function Documentation	1826
6.493.4.1 operator[]()	1827
6.493.4.2 size()	1827
6.493.5 Member Data Documentation	1827
6.493.5.1 lhs	1827
6.493.5.2 rhs	1827
6.494 Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > Class Template Reference	1828
6.494.1 Detailed Description	1829
6.494.2 Member Typedef Documentation	1829
6.494.2.1 LHS_EXT	1829
6.494.2.2 LHS_TYPE	1829
6.494.2.3 RHS_EXT	1829
6.494.2.4 RHS_TYPE	1830
6.494.2.5 STORAGE	1830
6.494.3 Constructor & Destructor Documentation	1830
6.494.3.1 Plus_Vector_Vector()	1830
6.494.4 Member Function Documentation	1830
6.494.4.1 operator[]()	1830
6.494.4.2 size()	1831
6.494.5 Member Data Documentation	1831
6.494.5.1 lhs	1831
6.494.5.2 rhs	1831
6.495 Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > Class Template Reference	1832
6.495.1 Detailed Description	1833
6.495.2 Member Typedef Documentation	1833
6.495.2.1 LHS_EXT	1833



6.495.2.2 LHS_TYPE . . . . .	1833
6.495.2.3 RHS_EXT . . . . .	1833
6.495.2.4 RHS_TYPE . . . . .	1834
6.495.2.5 STORAGE . . . . .	1834
6.495.3 Constructor & Destructor Documentation . . . . .	1834
6.495.3.1 Plus_Vector_Vector() . . . . .	1834
6.495.4 Member Function Documentation . . . . .	1834
6.495.4.1 operator[]() . . . . .	1834
6.495.4.2 size() . . . . .	1835
6.495.5 Member Data Documentation . . . . .	1835
6.495.5.1 lhs . . . . .	1835
6.495.5.2 rhs . . . . .	1835
6.496 Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > Class Template Reference . . . . .	1836
6.496.1 Detailed Description . . . . .	1837
6.496.2 Member Typedef Documentation . . . . .	1837
6.496.2.1 LHS_EXT . . . . .	1837
6.496.2.2 LHS_TYPE . . . . .	1837
6.496.2.3 RHS_EXT . . . . .	1837
6.496.2.4 RHS_TYPE . . . . .	1838
6.496.2.5 STORAGE . . . . .	1838
6.496.3 Constructor & Destructor Documentation . . . . .	1838
6.496.3.1 Plus_Vector_Vector() . . . . .	1838
6.496.4 Member Function Documentation . . . . .	1838
6.496.4.1 operator[]() . . . . .	1838
6.496.4.2 size() . . . . .	1839
6.496.5 Member Data Documentation . . . . .	1839
6.496.5.1 lhs . . . . .	1839
6.496.5.2 rhs . . . . .	1839
6.497 Rcpp::sugar::pmax_op< RTYPE, LHS_NA, RHS_NA > Struct Template Reference . . . . .	1839
6.497.1 Detailed Description . . . . .	1839
6.498 Rcpp::sugar::pmax_op< INTSXP, LHS_NA, RHS_NA > Struct Template Reference . . . . .	1840
6.498.1 Detailed Description . . . . .	1840
6.498.2 Member Function Documentation . . . . .	1840
6.498.2.1 operator>() . . . . .	1840
6.499 Rcpp::sugar::pmax_op< REALSXP, false, false > Struct Reference . . . . .	1840
6.499.1 Detailed Description . . . . .	1841
6.499.2 Member Function Documentation . . . . .	1841
6.499.2.1 operator>() . . . . .	1841
6.500 Rcpp::sugar::pmax_op< REALSXP, false, true > Struct Reference . . . . .	1841

---

6.500.1 Detailed Description	1841
6.500.2 Member Function Documentation	1841
6.500.2.1 operator>()	1842
6.501 Rcpp::sugar::pmax_op< REALSXP, true, false > Struct Reference	1842
6.501.1 Detailed Description	1842
6.501.2 Member Function Documentation	1842
6.501.2.1 operator>()	1842
6.502 Rcpp::sugar::pmax_op< REALSXP, true, true > Struct Reference	1843
6.502.1 Detailed Description	1843
6.502.2 Member Function Documentation	1843
6.502.2.1 operator>()	1843
6.503 Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA > Class Template Reference	1844
6.503.1 Detailed Description	1844
6.503.2 Member Typedef Documentation	1844
6.503.2.1 STORAGE	1844
6.503.3 Constructor & Destructor Documentation	1844
6.503.3.1 pmax_op_Vector_Primitive()	1845
6.503.4 Member Function Documentation	1845
6.503.4.1 operator>()	1845
6.503.5 Member Data Documentation	1845
6.503.5.1 right	1845
6.504 Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true > Class Reference	1845
6.504.1 Detailed Description	1846
6.504.2 Constructor & Destructor Documentation	1846
6.504.2.1 pmax_op_Vector_Primitive()	1846
6.504.3 Member Function Documentation	1846
6.504.3.1 operator>()	1846
6.504.4 Member Data Documentation	1847
6.504.4.1 right	1847
6.505 Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T > Class Template Reference	1847
6.505.1 Detailed Description	1848
6.505.2 Member Typedef Documentation	1848
6.505.2.1 OPERATOR	1849
6.505.2.2 STORAGE	1849
6.505.3 Constructor & Destructor Documentation	1849
6.505.3.1 Pmax_Vector_Primitive()	1849
6.505.4 Member Function Documentation	1849
6.505.4.1 operator[]()	1849
6.505.4.2 size()	1850

---

6.505.5 Member Data Documentation	1850
6.505.5.1 lhs	1850
6.505.5.2 op	1850
6.506 Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	1851
6.506.1 Detailed Description	1852
6.506.2 Member Typedef Documentation	1852
6.506.2.1 OPERATOR	1852
6.506.2.2 STORAGE	1852
6.506.3 Constructor & Destructor Documentation	1852
6.506.3.1 Pmax_Vector_Vector()	1852
6.506.4 Member Function Documentation	1853
6.506.4.1 operator[]()	1853
6.506.4.2 size()	1853
6.506.5 Member Data Documentation	1853
6.506.5.1 lhs	1853
6.506.5.2 op	1854
6.506.5.3 rhs	1854
6.507 Rcpp::sugar::pmin_op< RTYPE, LHS_NA, RHS_NA > Struct Template Reference	1854
6.507.1 Detailed Description	1854
6.508 Rcpp::sugar::pmin_op< INTSXP, LHS_NA, RHS_NA > Struct Template Reference	1854
6.508.1 Detailed Description	1855
6.508.2 Member Function Documentation	1855
6.508.2.1 operator()	1855
6.509 Rcpp::sugar::pmin_op< REALSXP, false, false > Struct Reference	1855
6.509.1 Detailed Description	1855
6.509.2 Member Function Documentation	1856
6.509.2.1 operator()	1856
6.510 Rcpp::sugar::pmin_op< REALSXP, false, true > Struct Reference	1856
6.510.1 Detailed Description	1856
6.510.2 Member Function Documentation	1856
6.510.2.1 operator()	1856
6.511 Rcpp::sugar::pmin_op< REALSXP, true, false > Struct Reference	1857
6.511.1 Detailed Description	1857
6.511.2 Member Function Documentation	1857
6.511.2.1 operator()	1857
6.512 Rcpp::sugar::pmin_op< REALSXP, true, true > Struct Reference	1857
6.512.1 Detailed Description	1857
6.512.2 Member Function Documentation	1858

6.512.2.1 operator>()	1858
6.513 Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA > Class Template Reference	1858
6.513.1 Detailed Description	1859
6.513.2 Member Typedef Documentation	1859
6.513.2.1 STORAGE	1859
6.513.3 Constructor & Destructor Documentation	1859
6.513.3.1 pmin_op_Vector_Primitive()	1859
6.513.4 Member Function Documentation	1859
6.513.4.1 operator>()	1860
6.513.5 Member Data Documentation	1860
6.513.5.1 right	1860
6.514 Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true > Class Reference	1860
6.514.1 Detailed Description	1861
6.514.2 Constructor & Destructor Documentation	1861
6.514.2.1 pmin_op_Vector_Primitive()	1861
6.514.3 Member Function Documentation	1861
6.514.3.1 operator>()	1861
6.514.4 Member Data Documentation	1861
6.514.4.1 right	1862
6.515 Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T > Class Template Reference	1862
6.515.1 Detailed Description	1863
6.515.2 Member Typedef Documentation	1863
6.515.2.1 OPERATOR	1863
6.515.2.2 STORAGE	1863
6.515.3 Constructor & Destructor Documentation	1864
6.515.3.1 Pmin_Vector_Primitive()	1864
6.515.4 Member Function Documentation	1864
6.515.4.1 operator[]()	1864
6.515.4.2 size()	1864
6.515.5 Member Data Documentation	1864
6.515.5.1 lhs	1865
6.515.5.2 op	1865
6.516 Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	1865
6.516.1 Detailed Description	1866
6.516.2 Member Typedef Documentation	1866
6.516.2.1 OPERATOR	1866
6.516.2.2 STORAGE	1866
6.516.3 Constructor & Destructor Documentation	1867

---

6.516.3.1 Pmin_Vector_Vector()	1867
6.516.4 Member Function Documentation	1867
6.516.4.1 operator[]()	1867
6.516.4.2 size()	1867
6.516.5 Member Data Documentation	1867
6.516.5.1 lhs	1868
6.516.5.2 op	1868
6.516.5.3 rhs	1868
6.517 Rcpp::traits::pointer_wrap_tag Struct Reference	1868
6.517.1 Detailed Description	1868
6.518 Rcpp::stats::PoissonGenerator Class Reference	1869
6.518.1 Detailed Description	1870
6.518.2 Constructor & Destructor Documentation	1870
6.518.2.1 PoissonGenerator()	1870
6.518.3 Member Function Documentation	1870
6.518.3.1 operator>()()	1870
6.518.4 Member Data Documentation	1870
6.518.4.1 mu	1870
6.519 Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE > Class Template Reference	1871
6.519.1 Detailed Description	1872
6.519.2 Member Typedef Documentation	1873
6.519.2.1 STORAGE	1873
6.519.3 Constructor & Destructor Documentation	1873
6.519.3.1 Pow()	1873
6.519.4 Member Function Documentation	1873
6.519.4.1 operator[]()	1873
6.519.4.2 size()	1874
6.519.5 Member Data Documentation	1874
6.519.5.1 object	1874
6.519.5.2 op	1874
6.520 Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE > Class Template Reference	1875
6.520.1 Detailed Description	1876
6.520.2 Constructor & Destructor Documentation	1876
6.520.2.1 Pow()	1877
6.520.3 Member Function Documentation	1877
6.520.3.1 operator[]()	1877
6.520.3.2 size()	1877
6.520.4 Member Data Documentation	1877
6.520.4.1 object	1878

---

6.520.4.2 op . . . . .	1878
6.521 Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE > Class Template Reference . . . . .	1878
6.521.1 Detailed Description . . . . .	1879
6.521.2 Constructor & Destructor Documentation . . . . .	1879
6.521.2.1 Pow() . . . . .	1880
6.521.3 Member Function Documentation . . . . .	1880
6.521.3.1 operator[]() . . . . .	1880
6.521.3.2 size() . . . . .	1880
6.521.4 Member Data Documentation . . . . .	1880
6.521.4.1 object . . . . .	1881
6.521.4.2 op . . . . .	1881
6.522 Rcpp::PreserveStorage< CLASS > Class Template Reference . . . . .	1881
6.522.1 Detailed Description . . . . .	1882
6.522.2 Constructor & Destructor Documentation . . . . .	1882
6.522.2.1 PreserveStorage() . . . . .	1882
6.522.2.2 ~PreserveStorage() . . . . .	1882
6.522.3 Member Function Documentation . . . . .	1882
6.522.3.1 copy__() . . . . .	1883
6.522.3.2 get__() . . . . .	1883
6.522.3.3 inherits() . . . . .	1883
6.522.3.4 invalidate__() . . . . .	1884
6.522.3.5 operator SEXP() . . . . .	1884
6.522.3.6 set__() . . . . .	1884
6.522.4 Member Data Documentation . . . . .	1885
6.522.4.1 data . . . . .	1885
6.522.4.2 token . . . . .	1885
6.523 Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy Class Reference . . . . .	1885
6.523.1 Detailed Description . . . . .	1886
6.523.2 Constructor & Destructor Documentation . . . . .	1886
6.523.2.1 ProtectedProxy() . . . . .	1887
6.523.3 Member Function Documentation . . . . .	1887
6.523.3.1 get() . . . . .	1887
6.523.3.2 operator SEXP() . . . . .	1887
6.523.3.3 operator U() . . . . .	1888
6.523.3.4 operator=() . . . . .	1888
6.523.3.5 set() . . . . .	1889
6.523.4 Member Data Documentation . . . . .	1889
6.523.4.1 xp . . . . .	1889
6.524 Rcpp::ProtectedProxyPolicy< XPtrClass > Class Template Reference . . . . .	1889

---

6.524.1 Detailed Description	1890
6.524.2 Member Function Documentation	1890
6.524.2.1 prot() [1/2]	1890
6.524.2.2 prot() [2/2]	1890
6.525 Rcpp::traits::proxy_based_const_iterator< RTYPE, StoragePolicy > Struct Template Reference	1890
6.525.1 Detailed Description	1891
6.525.2 Member Typedef Documentation	1891
6.525.2.1 type	1891
6.526 Rcpp::traits::proxy_based_iterator< RTYPE, StoragePolicy > Struct Template Reference	1891
6.526.1 Detailed Description	1891
6.526.2 Member Typedef Documentation	1892
6.526.2.1 type	1892
6.527 Rcpp::traits::proxy_cache< RTYPE, StoragePolicy > Class Template Reference	1892
6.527.1 Detailed Description	1893
6.527.2 Member Typedef Documentation	1893
6.527.2.1 const_iterator	1893
6.527.2.2 const_proxy	1893
6.527.2.3 iterator	1894
6.527.2.4 proxy	1894
6.527.2.5 VECTOR	1894
6.527.3 Constructor & Destructor Documentation	1894
6.527.3.1 proxy_cache()	1894
6.527.3.2 ~proxy_cache()	1894
6.527.4 Member Function Documentation	1895
6.527.4.1 get()	1895
6.527.4.2 get_const()	1895
6.527.4.3 ref() [1/4]	1895
6.527.4.4 ref() [2/4]	1895
6.527.4.5 ref() [3/4]	1896
6.527.4.6 ref() [4/4]	1896
6.527.4.7 update()	1896
6.527.5 Member Data Documentation	1896
6.527.5.1 p	1896
6.528 Rcpp::stats::Q0< RTYPE, NA, T > Class Template Reference	1897
6.528.1 Detailed Description	1898
6.528.2 Member Typedef Documentation	1898
6.528.2.1 FunPtr	1898
6.528.2.2 VEC_TYPE	1898
6.528.3 Constructor & Destructor Documentation	1898

6.528.3.1 Q0()	1899
6.528.4 Member Function Documentation	1899
6.528.4.1 operator[]()	1899
6.528.4.2 size()	1899
6.528.5 Member Data Documentation	1900
6.528.5.1 log	1900
6.528.5.2 lower	1900
6.528.5.3 ptr	1900
6.528.5.4 vec	1900
6.529 Rcpp::stats::Q1< RTYPE, NA, T > Class Template Reference	1901
6.529.1 Detailed Description	1902
6.529.2 Member Typedef Documentation	1902
6.529.2.1 FunPtr	1902
6.529.2.2 VEC_TYPE	1902
6.529.3 Constructor & Destructor Documentation	1902
6.529.3.1 Q1()	1903
6.529.4 Member Function Documentation	1903
6.529.4.1 operator[]()	1903
6.529.4.2 size()	1903
6.529.5 Member Data Documentation	1904
6.529.5.1 log	1904
6.529.5.2 lower	1904
6.529.5.3 p0	1904
6.529.5.4 ptr	1904
6.529.5.5 vec	1905
6.530 Rcpp::stats::Q2< RTYPE, NA, T > Class Template Reference	1905
6.530.1 Detailed Description	1906
6.530.2 Member Typedef Documentation	1906
6.530.2.1 FunPtr	1906
6.530.2.2 VEC_TYPE	1907
6.530.3 Constructor & Destructor Documentation	1907
6.530.3.1 Q2()	1907
6.530.4 Member Function Documentation	1907
6.530.4.1 operator[]()	1907
6.530.4.2 size()	1908
6.530.5 Member Data Documentation	1908
6.530.5.1 log	1908
6.530.5.2 lower	1908
6.530.5.3 p0	1909



6.530.5.4 p1 . . . . .	1909
6.530.5.5 ptr . . . . .	1909
6.530.5.6 vec . . . . .	1909
6.531 Rcpp::stats::Q3< RTYPE, NA, T > Class Template Reference . . . . .	1910
6.531.1 Detailed Description . . . . .	1911
6.531.2 Member Typedef Documentation . . . . .	1911
6.531.2.1 FunPtr . . . . .	1911
6.531.2.2 VEC_TYPE . . . . .	1911
6.531.3 Constructor & Destructor Documentation . . . . .	1911
6.531.3.1 Q3() . . . . .	1912
6.531.4 Member Function Documentation . . . . .	1912
6.531.4.1 operator[]() . . . . .	1912
6.531.4.2 size() . . . . .	1912
6.531.5 Member Data Documentation . . . . .	1913
6.531.5.1 log . . . . .	1913
6.531.5.2 lower . . . . .	1913
6.531.5.3 p0 . . . . .	1913
6.531.5.4 p1 . . . . .	1913
6.531.5.5 p2 . . . . .	1914
6.531.5.6 ptr . . . . .	1914
6.531.5.7 vec . . . . .	1914
6.532 Rcpp::traits::expands_to_logical__impl< LGLSXP >::r_expands_to_logical Struct Reference . . . . .	1914
6.532.1 Detailed Description . . . . .	1914
6.533 Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_matrix_interface Struct Reference . . . . .	1915
6.533.1 Detailed Description . . . . .	1915
6.534 Rcpp::traits::r_sexptype_needscastr< T > Struct Template Reference . . . . .	1915
6.534.1 Detailed Description . . . . .	1916
6.535 Rcpp::traits::r_sexptype_needscastr< double > Struct Reference . . . . .	1916
6.535.1 Detailed Description . . . . .	1917
6.536 Rcpp::traits::r_sexptype_needscastr< int > Struct Reference . . . . .	1917
6.536.1 Detailed Description . . . . .	1918
6.537 Rcpp::traits::r_sexptype_needscastr< Rbyte > Struct Reference . . . . .	1918
6.537.1 Detailed Description . . . . .	1919
6.538 Rcpp::traits::r_sexptype_needscastr< Rcomplex > Struct Reference . . . . .	1919
6.538.1 Detailed Description . . . . .	1920
6.539 Rcpp::traits::r_sexptype_traits< T > Struct Template Reference . . . . .	1920
6.539.1 Detailed Description . . . . .	1921
6.539.2 Member Enumeration Documentation . . . . .	1921
6.539.2.1 anonymous enum . . . . .	1921

6.540 Rcpp::traits::r_sexptype_traits< bool > Struct Reference	1921
6.540.1 Detailed Description	1922
6.540.2 Member Enumeration Documentation	1922
6.540.2.1 anonymous enum	1922
6.541 Rcpp::traits::r_sexptype_traits< const double > Struct Reference	1922
6.541.1 Detailed Description	1922
6.541.2 Member Enumeration Documentation	1922
6.541.2.1 anonymous enum	1922
6.542 Rcpp::traits::r_sexptype_traits< const int > Struct Reference	1923
6.542.1 Detailed Description	1923
6.542.2 Member Enumeration Documentation	1923
6.542.2.1 anonymous enum	1923
6.543 Rcpp::traits::r_sexptype_traits< double > Struct Reference	1924
6.543.1 Detailed Description	1924
6.543.2 Member Enumeration Documentation	1924
6.543.2.1 anonymous enum	1924
6.544 Rcpp::traits::r_sexptype_traits< float > Struct Reference	1924
6.544.1 Detailed Description	1925
6.544.2 Member Enumeration Documentation	1925
6.544.2.1 anonymous enum	1925
6.545 Rcpp::traits::r_sexptype_traits< int > Struct Reference	1925
6.545.1 Detailed Description	1925
6.545.2 Member Enumeration Documentation	1925
6.545.2.1 anonymous enum	1925
6.546 Rcpp::traits::r_sexptype_traits< long > Struct Reference	1926
6.546.1 Detailed Description	1926
6.546.2 Member Enumeration Documentation	1926
6.546.2.1 anonymous enum	1926
6.547 Rcpp::traits::r_sexptype_traits< long double > Struct Reference	1927
6.547.1 Detailed Description	1927
6.547.2 Member Enumeration Documentation	1927
6.547.2.1 anonymous enum	1927
6.548 Rcpp::traits::r_sexptype_traits< Rbyte > Struct Reference	1927
6.548.1 Detailed Description	1928
6.548.2 Member Enumeration Documentation	1928
6.548.2.1 anonymous enum	1928
6.549 Rcpp::traits::r_sexptype_traits< Rcomplex > Struct Reference	1928
6.549.1 Detailed Description	1928
6.549.2 Member Enumeration Documentation	1928

---

6.549.2.1 anonymous enum	1928
6.550 Rcpp::traits::r_sexptype_traits< Rcpp::Date > Struct Reference	1929
6.550.1 Detailed Description	1929
6.550.2 Member Enumeration Documentation	1929
6.550.2.1 anonymous enum	1929
6.551 Rcpp::traits::r_sexptype_traits< Rcpp::Datetime > Struct Reference	1930
6.551.1 Detailed Description	1930
6.551.2 Member Enumeration Documentation	1930
6.551.2.1 anonymous enum	1930
6.552 Rcpp::traits::r_sexptype_traits< Rcpp::String > Struct Reference	1930
6.552.1 Detailed Description	1931
6.552.2 Member Enumeration Documentation	1931
6.552.2.1 anonymous enum	1931
6.553 Rcpp::traits::r_sexptype_traits< short > Struct Reference	1931
6.553.1 Detailed Description	1931
6.553.2 Member Enumeration Documentation	1931
6.553.2.1 anonymous enum	1931
6.554 Rcpp::traits::r_sexptype_traits< std::complex< double > > Struct Reference	1932
6.554.1 Detailed Description	1932
6.554.2 Member Enumeration Documentation	1932
6.554.2.1 anonymous enum	1932
6.555 Rcpp::traits::r_sexptype_traits< std::complex< float > > Struct Reference	1933
6.555.1 Detailed Description	1933
6.555.2 Member Enumeration Documentation	1933
6.555.2.1 anonymous enum	1933
6.556 Rcpp::traits::r_sexptype_traits< std::string > Struct Reference	1933
6.556.1 Detailed Description	1934
6.556.2 Member Enumeration Documentation	1934
6.556.2.1 anonymous enum	1934
6.557 Rcpp::traits::r_sexptype_traits< unsigned int > Struct Reference	1934
6.557.1 Detailed Description	1934
6.557.2 Member Enumeration Documentation	1934
6.557.2.1 anonymous enum	1934
6.558 Rcpp::traits::r_sexptype_traits< unsigned long > Struct Reference	1935
6.558.1 Detailed Description	1935
6.558.2 Member Enumeration Documentation	1935
6.558.2.1 anonymous enum	1935
6.559 Rcpp::traits::r_sexptype_traits< unsigned short > Struct Reference	1936
6.559.1 Detailed Description	1936

---

6.559.2 Member Enumeration Documentation	1936
6.559.2.1 anonymous enum	1936
6.560 Rcpp::Matrix< RTYPE, StoragePolicy >::r_type Struct Reference	1936
6.560.1 Detailed Description	1937
6.561 Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_type Struct Reference	1938
6.561.1 Detailed Description	1938
6.562 Rcpp::VectorBase< RTYPE, na, VECTOR >::r_type Struct Reference	1939
6.562.1 Detailed Description	1939
6.563 Rcpp::traits::r_type_enum_tag Struct Reference	1940
6.563.1 Detailed Description	1940
6.564 Rcpp::traits::r_type_generic_tag Struct Reference	1940
6.564.1 Detailed Description	1940
6.565 Rcpp::traits::r_type_module_object_const_pointer_tag Struct Reference	1940
6.565.1 Detailed Description	1940
6.566 Rcpp::traits::r_type_module_object_const_reference_tag Struct Reference	1941
6.566.1 Detailed Description	1941
6.567 Rcpp::traits::r_type_module_object_pointer_tag Struct Reference	1941
6.567.1 Detailed Description	1941
6.568 Rcpp::traits::r_type_module_object_reference_tag Struct Reference	1941
6.568.1 Detailed Description	1941
6.569 Rcpp::traits::r_type_module_object_tag Struct Reference	1942
6.569.1 Detailed Description	1942
6.570 Rcpp::traits::r_type_pair_tag Struct Reference	1942
6.570.1 Detailed Description	1942
6.571 Rcpp::traits::r_type_pairstring_generic_tag Struct Reference	1942
6.571.1 Detailed Description	1942
6.572 Rcpp::traits::r_type_pairstring_primitive_tag Struct Reference	1943
6.572.1 Detailed Description	1943
6.573 Rcpp::traits::r_type_pairstring_string_tag Struct Reference	1943
6.573.1 Detailed Description	1943
6.574 Rcpp::traits::r_type_primitive_tag Struct Reference	1943
6.574.1 Detailed Description	1943
6.575 Rcpp::traits::r_type_RcppString_tag Struct Reference	1944
6.575.1 Detailed Description	1944
6.576 Rcpp::traits::r_type_string_tag Struct Reference	1944
6.576.1 Detailed Description	1944
6.577 Rcpp::traits::r_type_traits< T > Struct Template Reference	1944
6.577.1 Detailed Description	1944
6.577.2 Member Typedef Documentation	1945

---

6.577.2.1 r_category	1945
6.578 Rcpp::traits::r_type_traits< bool > Struct Reference	1945
6.578.1 Detailed Description	1945
6.578.2 Member Typedef Documentation	1945
6.578.2.1 r_category	1945
6.579 Rcpp::traits::r_type_traits< char > Struct Reference	1946
6.579.1 Detailed Description	1946
6.579.2 Member Typedef Documentation	1946
6.579.2.1 r_category	1946
6.580 Rcpp::traits::r_type_traits< const char * > Struct Reference	1946
6.580.1 Detailed Description	1946
6.580.2 Member Typedef Documentation	1947
6.580.2.1 r_category	1947
6.581 Rcpp::traits::r_type_traits< const double > Struct Reference	1947
6.581.1 Detailed Description	1947
6.581.2 Member Typedef Documentation	1947
6.581.2.1 r_category	1947
6.582 Rcpp::traits::r_type_traits< const int > Struct Reference	1948
6.582.1 Detailed Description	1948
6.582.2 Member Typedef Documentation	1948
6.582.2.1 r_category	1948
6.583 Rcpp::traits::r_type_traits< const wchar_t * > Struct Reference	1948
6.583.1 Detailed Description	1948
6.583.2 Member Typedef Documentation	1949
6.583.2.1 r_category	1949
6.584 Rcpp::traits::r_type_traits< double > Struct Reference	1949
6.584.1 Detailed Description	1949
6.584.2 Member Typedef Documentation	1949
6.584.2.1 r_category	1949
6.585 Rcpp::traits::r_type_traits< float > Struct Reference	1950
6.585.1 Detailed Description	1950
6.585.2 Member Typedef Documentation	1950
6.585.2.1 r_category	1950
6.586 Rcpp::traits::r_type_traits< int > Struct Reference	1950
6.586.1 Detailed Description	1950
6.586.2 Member Typedef Documentation	1951
6.586.2.1 r_category	1951
6.587 Rcpp::traits::r_type_traits< long > Struct Reference	1951
6.587.1 Detailed Description	1951

---

6.587.2 Member Typedef Documentation	1951
6.587.2.1 r_category	1951
6.588 Rcpp::traits::r_type_traits< long double > Struct Reference	1952
6.588.1 Detailed Description	1952
6.588.2 Member Typedef Documentation	1952
6.588.2.1 r_category	1952
6.589 Rcpp::traits::r_type_traits< Rbyte > Struct Reference	1952
6.589.1 Detailed Description	1952
6.589.2 Member Typedef Documentation	1953
6.589.2.1 r_category	1953
6.590 Rcpp::traits::r_type_traits< Rcomplex > Struct Reference	1953
6.590.1 Detailed Description	1953
6.590.2 Member Typedef Documentation	1953
6.590.2.1 r_category	1953
6.591 Rcpp::traits::r_type_traits< Rcpp::Date > Struct Reference	1954
6.591.1 Detailed Description	1954
6.591.2 Member Typedef Documentation	1954
6.591.2.1 r_category	1954
6.592 Rcpp::traits::r_type_traits< Rcpp::Datetime > Struct Reference	1954
6.592.1 Detailed Description	1954
6.592.2 Member Typedef Documentation	1955
6.592.2.1 r_category	1955
6.593 Rcpp::traits::r_type_traits< Rcpp::object< T > > Struct Template Reference	1955
6.593.1 Detailed Description	1955
6.593.2 Member Typedef Documentation	1955
6.593.2.1 r_category	1956
6.594 Rcpp::traits::r_type_traits< Rcpp::String > Struct Reference	1956
6.594.1 Detailed Description	1956
6.594.2 Member Typedef Documentation	1956
6.594.2.1 r_category	1956
6.595 Rcpp::traits::r_type_traits< short > Struct Reference	1956
6.595.1 Detailed Description	1957
6.595.2 Member Typedef Documentation	1957
6.595.2.1 r_category	1957
6.596 Rcpp::traits::r_type_traits< std::complex< double > > Struct Reference	1957
6.596.1 Detailed Description	1957
6.596.2 Member Typedef Documentation	1957
6.596.2.1 r_category	1958
6.597 Rcpp::traits::r_type_traits< std::complex< float > > Struct Reference	1958

---

6.597.1 Detailed Description . . . . .	1958
6.597.2 Member Typedef Documentation . . . . .	1958
6.597.2.1 r_category . . . . .	1958
6.598 Rcpp::traits::r_type_traits< std::pair< const KEY, VALUE > > Struct Template Reference . . . . .	1958
6.598.1 Detailed Description . . . . .	1959
6.598.2 Member Typedef Documentation . . . . .	1959
6.598.2.1 r_category . . . . .	1959
6.599 Rcpp::traits::r_type_traits< std::pair< const std::string, bool > > Struct Reference . . . . .	1959
6.599.1 Detailed Description . . . . .	1959
6.599.2 Member Typedef Documentation . . . . .	1960
6.599.2.1 r_category . . . . .	1960
6.600 Rcpp::traits::r_type_traits< std::pair< const std::string, char > > Struct Reference . . . . .	1960
6.600.1 Detailed Description . . . . .	1960
6.600.2 Member Typedef Documentation . . . . .	1960
6.600.2.1 r_category . . . . .	1960
6.601 Rcpp::traits::r_type_traits< std::pair< const std::string, const int > > Struct Reference . . . . .	1961
6.601.1 Detailed Description . . . . .	1961
6.601.2 Member Typedef Documentation . . . . .	1961
6.601.2.1 r_category . . . . .	1961
6.602 Rcpp::traits::r_type_traits< std::pair< const std::string, double > > Struct Reference . . . . .	1961
6.602.1 Detailed Description . . . . .	1961
6.602.2 Member Typedef Documentation . . . . .	1962
6.602.2.1 r_category . . . . .	1962
6.603 Rcpp::traits::r_type_traits< std::pair< const std::string, float > > Struct Reference . . . . .	1962
6.603.1 Detailed Description . . . . .	1962
6.603.2 Member Typedef Documentation . . . . .	1962
6.603.2.1 r_category . . . . .	1962
6.604 Rcpp::traits::r_type_traits< std::pair< const std::string, int > > Struct Reference . . . . .	1963
6.604.1 Detailed Description . . . . .	1963
6.604.2 Member Typedef Documentation . . . . .	1963
6.604.2.1 r_category . . . . .	1963
6.605 Rcpp::traits::r_type_traits< std::pair< const std::string, long > > Struct Reference . . . . .	1963
6.605.1 Detailed Description . . . . .	1963
6.605.2 Member Typedef Documentation . . . . .	1964
6.605.2.1 r_category . . . . .	1964
6.606 Rcpp::traits::r_type_traits< std::pair< const std::string, long double > > Struct Reference . . . . .	1964
6.606.1 Detailed Description . . . . .	1964
6.606.2 Member Typedef Documentation . . . . .	1964
6.606.2.1 r_category . . . . .	1964

---

6.607 Rcpp::traits::r_type_traits< std::pair< const std::string, Rbyte > > Struct Reference . . . . .	1965
6.607.1 Detailed Description . . . . .	1965
6.607.2 Member Typedef Documentation . . . . .	1965
6.607.2.1 r_category . . . . .	1965
6.608 Rcpp::traits::r_type_traits< std::pair< const std::string, Rcomplex > > Struct Reference . . . . .	1965
6.608.1 Detailed Description . . . . .	1965
6.608.2 Member Typedef Documentation . . . . .	1966
6.608.2.1 r_category . . . . .	1966
6.609 Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Date > > Struct Reference . . . . .	1966
6.609.1 Detailed Description . . . . .	1966
6.609.2 Member Typedef Documentation . . . . .	1966
6.609.2.1 r_category . . . . .	1966
6.610 Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Datetime > > Struct Reference . . . . .	1967
6.610.1 Detailed Description . . . . .	1967
6.610.2 Member Typedef Documentation . . . . .	1967
6.610.2.1 r_category . . . . .	1967
6.611 Rcpp::traits::r_type_traits< std::pair< const std::string, short > > Struct Reference . . . . .	1967
6.611.1 Detailed Description . . . . .	1967
6.611.2 Member Typedef Documentation . . . . .	1968
6.611.2.1 r_category . . . . .	1968
6.612 Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex< double > > > Struct Reference . . . . .	1968
6.612.1 Detailed Description . . . . .	1968
6.612.2 Member Typedef Documentation . . . . .	1968
6.612.2.1 r_category . . . . .	1968
6.613 Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex< float > > > Struct Reference . . . . .	1969
6.613.1 Detailed Description . . . . .	1969
6.613.2 Member Typedef Documentation . . . . .	1969
6.613.2.1 r_category . . . . .	1969
6.614 Rcpp::traits::r_type_traits< std::pair< const std::string, std::string > > Struct Reference . . . . .	1969
6.614.1 Detailed Description . . . . .	1969
6.614.2 Member Typedef Documentation . . . . .	1970
6.614.2.1 r_category . . . . .	1970
6.615 Rcpp::traits::r_type_traits< std::pair< const std::string, std::wstring > > Struct Reference . . . . .	1970
6.615.1 Detailed Description . . . . .	1970
6.615.2 Member Typedef Documentation . . . . .	1970
6.615.2.1 r_category . . . . .	1970
6.616 Rcpp::traits::r_type_traits< std::pair< const std::string, T > > Struct Template Reference . . . . .	1971
6.616.1 Detailed Description . . . . .	1971
6.616.2 Member Typedef Documentation . . . . .	1971



---

6.616.2.1 <code>r_category</code> . . . . .	1971
6.617 <code>Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, unsigned int &gt; &gt;</code> Struct Reference . . . . .	1971
6.617.1 Detailed Description . . . . .	1972
6.617.2 Member Typedef Documentation . . . . .	1972
6.617.2.1 <code>r_category</code> . . . . .	1972
6.618 <code>Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, unsigned long &gt; &gt;</code> Struct Reference . . . . .	1972
6.618.1 Detailed Description . . . . .	1972
6.618.2 Member Typedef Documentation . . . . .	1972
6.618.2.1 <code>r_category</code> . . . . .	1973
6.619 <code>Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, unsigned short &gt; &gt;</code> Struct Reference . . . . .	1973
6.619.1 Detailed Description . . . . .	1973
6.619.2 Member Typedef Documentation . . . . .	1973
6.619.2.1 <code>r_category</code> . . . . .	1973
6.620 <code>Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, wchar_t &gt; &gt;</code> Struct Reference . . . . .	1974
6.620.1 Detailed Description . . . . .	1974
6.620.2 Member Typedef Documentation . . . . .	1974
6.620.2.1 <code>r_category</code> . . . . .	1974
6.621 <code>Rcpp::traits::r_type_traits&lt; std::string &gt;</code> Struct Reference . . . . .	1974
6.621.1 Detailed Description . . . . .	1974
6.621.2 Member Typedef Documentation . . . . .	1975
6.621.2.1 <code>r_category</code> . . . . .	1975
6.622 <code>Rcpp::traits::r_type_traits&lt; std::wstring &gt;</code> Struct Reference . . . . .	1975
6.622.1 Detailed Description . . . . .	1975
6.622.2 Member Typedef Documentation . . . . .	1975
6.622.2.1 <code>r_category</code> . . . . .	1975
6.623 <code>Rcpp::traits::r_type_traits&lt; unsigned int &gt;</code> Struct Reference . . . . .	1976
6.623.1 Detailed Description . . . . .	1976
6.623.2 Member Typedef Documentation . . . . .	1976
6.623.2.1 <code>r_category</code> . . . . .	1976
6.624 <code>Rcpp::traits::r_type_traits&lt; unsigned long &gt;</code> Struct Reference . . . . .	1976
6.624.1 Detailed Description . . . . .	1976
6.624.2 Member Typedef Documentation . . . . .	1977
6.624.2.1 <code>r_category</code> . . . . .	1977
6.625 <code>Rcpp::traits::r_type_traits&lt; unsigned short &gt;</code> Struct Reference . . . . .	1977
6.625.1 Detailed Description . . . . .	1977
6.625.2 Member Typedef Documentation . . . . .	1977
6.625.2.1 <code>r_category</code> . . . . .	1977
6.626 <code>Rcpp::traits::r_type_traits&lt; wchar_t &gt;</code> Struct Reference . . . . .	1978
6.626.1 Detailed Description . . . . .	1978

---

6.626.2 Member Typedef Documentation . . . . .	1978
6.626.2.1 r_category . . . . .	1978
6.627 Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy > Class Template Reference . . . . .	1978
6.627.1 Detailed Description . . . . .	1979
6.627.2 Member Typedef Documentation . . . . .	1979
6.627.2.1 const_iterator . . . . .	1979
6.627.2.2 const_proxy . . . . .	1980
6.627.2.3 iterator . . . . .	1980
6.627.2.4 proxy . . . . .	1980
6.627.2.5 storage_type . . . . .	1980
6.627.2.6 VECTOR . . . . .	1980
6.627.3 Constructor & Destructor Documentation . . . . .	1981
6.627.3.1 r_vector_cache() . . . . .	1981
6.627.4 Member Function Documentation . . . . .	1981
6.627.4.1 get() . . . . .	1981
6.627.4.2 get_const() . . . . .	1981
6.627.4.3 ref() [1/4] . . . . .	1981
6.627.4.4 ref() [2/4] . . . . .	1982
6.627.4.5 ref() [3/4] . . . . .	1982
6.627.4.6 ref() [4/4] . . . . .	1982
6.627.4.7 update() . . . . .	1982
6.627.5 Member Data Documentation . . . . .	1982
6.627.5.1 start . . . . .	1983
6.628 Rcpp::traits::r_vector_cache_type< RTYPE, StoragePolicy > Struct Template Reference . . . . .	1983
6.628.1 Detailed Description . . . . .	1983
6.628.2 Member Typedef Documentation . . . . .	1983
6.628.2.1 type . . . . .	1983
6.629 Rcpp::traits::r_vector_cache_type< EXPRXP, StoragePolicy > Struct Template Reference . . . . .	1984
6.629.1 Detailed Description . . . . .	1984
6.629.2 Member Typedef Documentation . . . . .	1984
6.629.2.1 type . . . . .	1984
6.630 Rcpp::traits::r_vector_cache_type< STRSXP, StoragePolicy > Struct Template Reference . . . . .	1984
6.630.1 Detailed Description . . . . .	1985
6.630.2 Member Typedef Documentation . . . . .	1985
6.630.2.1 type . . . . .	1985
6.631 Rcpp::traits::r_vector_cache_type< VECSXP, StoragePolicy > Struct Template Reference . . . . .	1985
6.631.1 Detailed Description . . . . .	1985
6.631.2 Member Typedef Documentation . . . . .	1986
6.631.2.1 type . . . . .	1986

---

6.632 Rcpp::traits::r_vector_const_iterator< RTYPE, StoragePolicy > Struct Template Reference . . . . .	1986
6.632.1 Detailed Description . . . . .	1986
6.632.2 Member Typedef Documentation . . . . .	1986
6.632.2.1 type . . . . .	1987
6.633 Rcpp::traits::r_vector_const_iterator< EXPRXP, StoragePolicy > Struct Template Reference . . . . .	1987
6.633.1 Detailed Description . . . . .	1988
6.634 Rcpp::traits::r_vector_const_iterator< STRXP, StoragePolicy > Struct Template Reference . . . . .	1988
6.634.1 Detailed Description . . . . .	1989
6.635 Rcpp::traits::r_vector_const_iterator< VECSXP, StoragePolicy > Struct Template Reference . . . . .	1990
6.635.1 Detailed Description . . . . .	1990
6.636 Rcpp::traits::r_vector_const_proxy< RTYPE, StoragePolicy > Struct Template Reference . . . . .	1991
6.636.1 Detailed Description . . . . .	1991
6.636.2 Member Typedef Documentation . . . . .	1991
6.636.2.1 type . . . . .	1991
6.637 Rcpp::traits::r_vector_const_proxy< EXPRXP, StoragePolicy > Struct Template Reference . . . . .	1992
6.637.1 Detailed Description . . . . .	1992
6.637.2 Member Typedef Documentation . . . . .	1992
6.637.2.1 type . . . . .	1992
6.638 Rcpp::traits::r_vector_const_proxy< STRXP, StoragePolicy > Struct Template Reference . . . . .	1992
6.638.1 Detailed Description . . . . .	1993
6.638.2 Member Typedef Documentation . . . . .	1993
6.638.2.1 type . . . . .	1993
6.639 Rcpp::traits::r_vector_const_proxy< VECSXP, StoragePolicy > Struct Template Reference . . . . .	1993
6.639.1 Detailed Description . . . . .	1993
6.639.2 Member Typedef Documentation . . . . .	1994
6.639.2.1 type . . . . .	1994
6.640 Rcpp::traits::r_vector_element_converter< RTYPE > Struct Template Reference . . . . .	1994
6.640.1 Detailed Description . . . . .	1994
6.640.2 Member Typedef Documentation . . . . .	1994
6.640.2.1 type . . . . .	1995
6.641 Rcpp::traits::r_vector_element_converter< EXPRXP > Struct Reference . . . . .	1995
6.641.1 Detailed Description . . . . .	1995
6.641.2 Member Typedef Documentation . . . . .	1995
6.641.2.1 type . . . . .	1995
6.642 Rcpp::traits::r_vector_element_converter< STRXP > Struct Reference . . . . .	1996
6.642.1 Detailed Description . . . . .	1996
6.642.2 Member Typedef Documentation . . . . .	1996
6.642.2.1 type . . . . .	1996
6.643 Rcpp::traits::r_vector_element_converter< VECSXP > Struct Reference . . . . .	1996

---

6.643.1 Detailed Description	1996
6.643.2 Member Typedef Documentation	1997
6.643.2.1 type	1997
6.644 Rcpp::traits::r_vector_iterator< RTYPE, StoragePolicy > Struct Template Reference	1997
6.644.1 Detailed Description	1997
6.644.2 Member Typedef Documentation	1997
6.644.2.1 type	1997
6.645 Rcpp::traits::r_vector_iterator< EXPRXP, StoragePolicy > Struct Template Reference	1998
6.645.1 Detailed Description	1998
6.646 Rcpp::traits::r_vector_iterator< STRXP, StoragePolicy > Struct Template Reference	1999
6.646.1 Detailed Description	2000
6.647 Rcpp::traits::r_vector_iterator< VECSXP, StoragePolicy > Struct Template Reference	2000
6.647.1 Detailed Description	2001
6.648 Rcpp::traits::r_vector_name_proxy< RTYPE, StoragePolicy > Struct Template Reference	2001
6.648.1 Detailed Description	2002
6.648.2 Member Typedef Documentation	2002
6.648.2.1 type	2002
6.649 Rcpp::traits::r_vector_name_proxy< EXPRXP, StoragePolicy > Struct Template Reference	2002
6.649.1 Detailed Description	2002
6.649.2 Member Typedef Documentation	2003
6.649.2.1 type	2003
6.650 Rcpp::traits::r_vector_name_proxy< STRXP, StoragePolicy > Struct Template Reference	2003
6.650.1 Detailed Description	2003
6.650.2 Member Typedef Documentation	2003
6.650.2.1 type	2004
6.651 Rcpp::traits::r_vector_name_proxy< VECSXP, StoragePolicy > Struct Template Reference	2004
6.651.1 Detailed Description	2004
6.651.2 Member Typedef Documentation	2004
6.651.2.1 type	2004
6.652 Rcpp::traits::r_vector_proxy< RTYPE, StoragePolicy > Struct Template Reference	2005
6.652.1 Detailed Description	2005
6.652.2 Member Typedef Documentation	2005
6.652.2.1 type	2005
6.653 Rcpp::traits::r_vector_proxy< EXPRXP, StoragePolicy > Struct Template Reference	2005
6.653.1 Detailed Description	2006
6.653.2 Member Typedef Documentation	2006
6.653.2.1 type	2006
6.654 Rcpp::traits::r_vector_proxy< STRXP, StoragePolicy > Struct Template Reference	2006
6.654.1 Detailed Description	2006

---

6.654.2 Member Typedef Documentation	2007
6.654.2.1 type	2007
6.655 Rcpp::traits::r_vector_proxy< VECSXP, StoragePolicy > Struct Template Reference	2007
6.655.1 Detailed Description	2007
6.655.2 Member Typedef Documentation	2007
6.655.2.1 type	2008
6.656 Rcpp::Range Class Reference	2008
6.656.1 Detailed Description	2009
6.656.2 Constructor & Destructor Documentation	2010
6.656.2.1 Range()	2010
6.656.3 Member Function Documentation	2010
6.656.3.1 get_end()	2010
6.656.3.2 get_start()	2010
6.656.3.3 operator+()	2011
6.656.3.4 operator++() [1/2]	2011
6.656.3.5 operator++() [2/2]	2011
6.656.3.6 operator+=()	2011
6.656.3.7 operator-()	2012
6.656.3.8 operator--() [1/2]	2012
6.656.3.9 operator--() [2/2]	2012
6.656.3.10 operator-=()	2012
6.656.3.11 operator[]()	2013
6.656.3.12 size()	2013
6.656.4 Member Data Documentation	2013
6.656.4.1 end_	2013
6.656.4.2 start	2013
6.657 Rcpp::sugar::Range< RTYPE, NA, T > Class Template Reference	2014
6.657.1 Detailed Description	2014
6.657.2 Member Typedef Documentation	2014
6.657.2.1 STORAGE	2014
6.657.3 Constructor & Destructor Documentation	2014
6.657.3.1 Range()	2015
6.657.4 Member Function Documentation	2015
6.657.4.1 operator Vector< RTYPE >()	2015
6.657.5 Member Data Documentation	2015
6.657.5.1 current	2015
6.657.5.2 max_	2016
6.657.5.3 min_	2016
6.657.5.4 obj	2016

---

6.658 Rcpp::sugar::Range< RTYPE, false, T > Class Template Reference . . . . .	2016
6.658.1 Detailed Description . . . . .	2017
6.658.2 Member Typedef Documentation . . . . .	2017
6.658.2.1 STORAGE . . . . .	2017
6.658.3 Constructor & Destructor Documentation . . . . .	2017
6.658.3.1 Range() . . . . .	2017
6.658.4 Member Function Documentation . . . . .	2018
6.658.4.1 operator Vector< RTYPE >() . . . . .	2018
6.658.5 Member Data Documentation . . . . .	2018
6.658.5.1 current . . . . .	2018
6.658.5.2 max_ . . . . .	2018
6.658.5.3 min_ . . . . .	2019
6.658.5.4 obj . . . . .	2019
6.659 Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR > Class Template Reference . . . . .	2019
6.659.1 Detailed Description . . . . .	2021
6.659.2 Member Typedef Documentation . . . . .	2021
6.659.2.1 iterator . . . . .	2021
6.659.2.2 Proxy . . . . .	2021
6.659.3 Constructor & Destructor Documentation . . . . .	2021
6.659.3.1 RangeIndexer() . . . . .	2021
6.659.4 Member Function Documentation . . . . .	2021
6.659.4.1 operator*=( ) . . . . .	2022
6.659.4.2 operator+=( ) . . . . .	2022
6.659.4.3 operator-=( ) . . . . .	2022
6.659.4.4 operator/=( ) . . . . .	2022
6.659.4.5 operator=( ) . . . . .	2023
6.659.4.6 operator[]( ) . . . . .	2023
6.659.4.7 size() . . . . .	2023
6.659.5 Member Data Documentation . . . . .	2023
6.659.5.1 size_ . . . . .	2023
6.659.5.2 start . . . . .	2024
6.660 Rcpp::VectorBase< RTYPE, na, VECTOR >::rcpp_sugar_expression Struct Reference . . . . .	2024
6.660.1 Detailed Description . . . . .	2024
6.661 Rcpp::ReferenceInputParameter< T > Class Template Reference . . . . .	2024
6.661.1 Detailed Description . . . . .	2025
6.661.2 Member Typedef Documentation . . . . .	2025
6.661.2.1 reference . . . . .	2025
6.661.3 Constructor & Destructor Documentation . . . . .	2025
6.661.3.1 ReferenceInputParameter() . . . . .	2025

---

6.661.4 Member Function Documentation	2025
6.661.4.1 operator reference()	2026
6.661.5 Member Data Documentation	2026
6.661.5.1 obj	2026
6.662 Rcpp::traits::remove_const< _Tp > Struct Template Reference	2026
6.662.1 Detailed Description	2026
6.662.2 Member Typedef Documentation	2027
6.662.2.1 type	2027
6.663 Rcpp::traits::remove_const< _Tp const > Struct Template Reference	2027
6.663.1 Detailed Description	2027
6.663.2 Member Typedef Documentation	2027
6.663.2.1 type	2027
6.664 Rcpp::traits::remove_const_and_reference< T > Struct Template Reference	2028
6.664.1 Detailed Description	2028
6.664.2 Member Typedef Documentation	2028
6.664.2.1 type	2028
6.665 Rcpp::traits::remove_reference< _Tp > Struct Template Reference	2028
6.665.1 Detailed Description	2029
6.665.2 Member Typedef Documentation	2029
6.665.2.1 type	2029
6.666 Rcpp::traits::remove_reference< _Tp & > Struct Template Reference	2029
6.666.1 Detailed Description	2029
6.666.2 Member Typedef Documentation	2029
6.666.2.1 type	2030
6.667 Rcpp::sugar::RemoveFromSet< SET > Class Template Reference	2030
6.667.1 Detailed Description	2030
6.667.2 Constructor & Destructor Documentation	2030
6.667.2.1 RemoveFromSet()	2030
6.667.3 Member Function Documentation	2031
6.667.3.1 operator>()	2031
6.667.4 Member Data Documentation	2031
6.667.4.1 set	2031
6.668 Rcpp::sugar::Rep< RTYPE, NA, T > Class Template Reference	2032
6.668.1 Detailed Description	2033
6.668.2 Member Typedef Documentation	2033
6.668.2.1 STORAGE	2033
6.668.2.2 VEC_TYPE	2033
6.668.3 Constructor & Destructor Documentation	2033
6.668.3.1 Rep()	2033

6.668.4 Member Function Documentation	2034
6.668.4.1 operator[]()	2034
6.668.4.2 size()	2034
6.668.5 Member Data Documentation	2034
6.668.5.1 n	2034
6.668.5.2 object	2034
6.668.5.3 times	2035
6.669 Rcpp::sugar::Rep_each< RTYPE, NA, T > Class Template Reference	2035
6.669.1 Detailed Description	2036
6.669.2 Member Typedef Documentation	2036
6.669.2.1 STORAGE	2036
6.669.2.2 VEC_TYPE	2036
6.669.3 Constructor & Destructor Documentation	2037
6.669.3.1 Rep_each()	2037
6.669.4 Member Function Documentation	2037
6.669.4.1 operator[]()	2037
6.669.4.2 size()	2037
6.669.5 Member Data Documentation	2037
6.669.5.1 n	2038
6.669.5.2 object	2038
6.669.5.3 times	2038
6.670 Rcpp::sugar::Rep_len< RTYPE, NA, T > Class Template Reference	2039
6.670.1 Detailed Description	2040
6.670.2 Member Typedef Documentation	2040
6.670.2.1 STORAGE	2040
6.670.2.2 VEC_TYPE	2040
6.670.3 Constructor & Destructor Documentation	2040
6.670.3.1 Rep_len()	2040
6.670.4 Member Function Documentation	2041
6.670.4.1 operator[]()	2041
6.670.4.2 size()	2041
6.670.5 Member Data Documentation	2041
6.670.5.1 len	2041
6.670.5.2 n	2042
6.670.5.3 object	2042
6.671 Rcpp::sugar::Rep_Single< T > Class Template Reference	2042
6.671.1 Detailed Description	2043
6.671.2 Constructor & Destructor Documentation	2043
6.671.2.1 Rep_Single()	2043



---

6.671.3 Member Function Documentation	2043
6.671.3.1 operator[]()	2044
6.671.3.2 size()	2044
6.671.4 Member Data Documentation	2044
6.671.4.1 n	2044
6.671.4.2 x	2044
6.672 Rcpp::result< T > Class Template Reference	2045
6.672.1 Detailed Description	2045
6.672.2 Constructor & Destructor Documentation	2045
6.672.2.1 result()	2045
6.672.3 Member Function Documentation	2045
6.672.3.1 operator T*()	2045
6.672.4 Member Data Documentation	2046
6.672.4.1 ptr	2046
6.673 Rcpp::sugar::median_detail::result< RTYPE > Struct Template Reference	2046
6.673.1 Detailed Description	2046
6.673.2 Member Typedef Documentation	2046
6.673.2.1 type	2047
6.673.3 Member Enumeration Documentation	2047
6.673.3.1 anonymous enum	2047
6.674 Rcpp::sugar::median_detail::result< INTSXP > Struct Reference	2047
6.674.1 Detailed Description	2047
6.674.2 Member Typedef Documentation	2048
6.674.2.1 type	2048
6.674.3 Member Enumeration Documentation	2048
6.674.3.1 anonymous enum	2048
6.675 Rcpp::sugar::median_detail::result< STRSXP > Struct Reference	2048
6.675.1 Detailed Description	2048
6.675.2 Member Typedef Documentation	2049
6.675.2.1 type	2049
6.675.3 Member Enumeration Documentation	2049
6.675.3.1 anonymous enum	2049
6.676 Rcpp::traits::result_of< T > Struct Template Reference	2049
6.676.1 Detailed Description	2049
6.676.2 Member Typedef Documentation	2050
6.676.2.1 type	2050
6.677 Rcpp::traits::result_of< RESULT_TYPE(*) (INPUT_TYPE) > Struct Template Reference	2050
6.677.1 Detailed Description	2050
6.677.2 Member Typedef Documentation	2050

---

6.677.2.1 type	2050
6.678 Rcpp::traits::result_of< RESULT_TYPE(*) (U1, U2) > Struct Template Reference	2051
6.678.1 Detailed Description	2051
6.678.2 Member Typedef Documentation	2051
6.678.2.1 type	2051
6.679 Rcpp::traits::result_of< RESULT_TYPE(*) (U1, U2, U3) > Struct Template Reference	2051
6.679.1 Detailed Description	2052
6.679.2 Member Typedef Documentation	2052
6.679.2.1 type	2052
6.680 Rcpp::sugar::Rev< RTYPE, NA, T > Class Template Reference	2052
6.680.1 Detailed Description	2053
6.680.2 Member Typedef Documentation	2053
6.680.2.1 STORAGE	2053
6.680.2.2 VEC_TYPE	2054
6.680.3 Constructor & Destructor Documentation	2054
6.680.3.1 Rev()	2054
6.680.4 Member Function Documentation	2054
6.680.4.1 operator[]()	2054
6.680.4.2 size()	2054
6.680.5 Member Data Documentation	2055
6.680.5.1 n	2055
6.680.5.2 object	2055
6.681 Rcpp::attributes::REExportsGenerator Class Reference	2055
6.681.1 Detailed Description	2056
6.681.2 Constructor & Destructor Documentation	2056
6.681.2.1 REExportsGenerator()	2057
6.681.3 Member Function Documentation	2057
6.681.3.1 commit()	2057
6.681.3.2 doWriteFunctions()	2058
6.681.3.3 writeBegin()	2058
6.681.3.4 writeEnd()	2059
6.681.4 Member Data Documentation	2059
6.681.4.1 registration_	2059
6.682 Rcpp::RNGScope Class Reference	2059
6.682.1 Detailed Description	2060
6.682.2 Constructor & Destructor Documentation	2060
6.682.2.1 RNGScope()	2060
6.682.2.2 ~RNGScope()	2060
6.683 Rcpp::RObjectMethods< Class > Class Template Reference	2061

---

6.683.1 Detailed Description	2061
6.683.2 Member Function Documentation	2061
6.683.2.1 isNULL()	2062
6.683.2.2 isObject()	2062
6.683.2.3 isS4()	2062
6.683.2.4 sexp_type()	2062
6.684 Rcpp::Rostream< OUTPUT > Class Template Reference	2063
6.684.1 Detailed Description	2064
6.684.2 Member Typedef Documentation	2064
6.684.2.1 Buffer	2064
6.684.3 Constructor & Destructor Documentation	2064
6.684.3.1 Rostream()	2064
6.684.4 Member Data Documentation	2064
6.684.4.1 buf	2064
6.685 Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T > Class Template Reference	2065
6.685.1 Detailed Description	2066
6.685.2 Member Typedef Documentation	2067
6.685.2.1 LHS_TYPE	2067
6.685.3 Constructor & Destructor Documentation	2067
6.685.3.1 Row()	2067
6.685.4 Member Function Documentation	2067
6.685.4.1 ncol()	2067
6.685.4.2 nrow()	2068
6.685.4.3 operator>()()	2068
6.685.4.4 size()	2068
6.685.5 Member Data Documentation	2068
6.685.5.1 nc	2068
6.685.5.2 nr	2069
6.686 Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM > Class Template Reference	2069
6.686.1 Detailed Description	2070
6.686.2 Member Typedef Documentation	2070
6.686.2.1 return_traits	2070
6.686.2.2 return_vector	2070
6.686.2.3 stored_type	2071
6.686.3 Constructor & Destructor Documentation	2071
6.686.3.1 RowMeansImpl()	2071
6.686.4 Member Function Documentation	2071
6.686.4.1 get()	2071
6.686.5 Member Data Documentation	2072

---

6.686.5.1 ref	2072
6.687 Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA_RM > Class Template Reference	2072
6.687.1 Detailed Description	2073
6.688 Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true > Class Template Reference	2073
6.688.1 Detailed Description	2074
6.688.2 Member Typedef Documentation	2075
6.688.2.1 return_traits	2075
6.688.2.2 return_vector	2075
6.688.2.3 stored_type	2075
6.688.3 Constructor & Destructor Documentation	2075
6.688.3.1 RowMeansImpl()	2075
6.688.4 Member Function Documentation	2076
6.688.4.1 get()	2076
6.688.5 Member Data Documentation	2076
6.688.5.1 ref	2077
6.689 Rcpp::sugar::detail::RowMeansReturn< RTYPE > Struct Template Reference	2077
6.689.1 Detailed Description	2077
6.689.2 Member Typedef Documentation	2078
6.689.2.1 type	2078
6.689.3 Member Enumeration Documentation	2078
6.689.3.1 anonymous enum	2078
6.690 Rcpp::sugar::detail::RowMeansReturn< CPLXSXP > Struct Reference	2078
6.690.1 Detailed Description	2079
6.690.2 Member Typedef Documentation	2079
6.690.2.1 type	2079
6.690.3 Member Enumeration Documentation	2079
6.690.3.1 anonymous enum	2079
6.691 Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM > Class Template Reference	2079
6.691.1 Detailed Description	2080
6.691.2 Member Typedef Documentation	2081
6.691.2.1 return_traits	2081
6.691.2.2 return_vector	2081
6.691.2.3 stored_type	2081
6.691.3 Constructor & Destructor Documentation	2081
6.691.3.1 RowSumsImpl()	2081
6.691.4 Member Function Documentation	2082
6.691.4.1 get()	2082
6.691.5 Member Data Documentation	2082
6.691.5.1 ref	2082

---

6.692 Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA_RM > Class Template Reference . . . . .	2083
6.692.1 Detailed Description . . . . .	2084
6.693 Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true > Class Template Reference . . . . .	2084
6.693.1 Detailed Description . . . . .	2085
6.693.2 Member Typedef Documentation . . . . .	2085
6.693.2.1 return_traits . . . . .	2085
6.693.2.2 return_vector . . . . .	2085
6.693.2.3 stored_type . . . . .	2086
6.693.3 Constructor & Destructor Documentation . . . . .	2086
6.693.3.1 RowSumsImpl() . . . . .	2086
6.693.4 Member Function Documentation . . . . .	2086
6.693.4.1 get() . . . . .	2086
6.693.5 Member Data Documentation . . . . .	2087
6.693.5.1 ref . . . . .	2087
6.694 Rcpp::sugar::detail::RowSumsReturn< RTYPE > Struct Template Reference . . . . .	2087
6.694.1 Detailed Description . . . . .	2088
6.694.2 Member Typedef Documentation . . . . .	2088
6.694.2.1 type . . . . .	2088
6.694.3 Member Enumeration Documentation . . . . .	2088
6.694.3.1 anonymous enum . . . . .	2088
6.695 Rcpp::sugar::detail::RowSumsReturn< LGLSXP > Struct Reference . . . . .	2089
6.695.1 Detailed Description . . . . .	2089
6.695.2 Member Typedef Documentation . . . . .	2089
6.695.2.1 type . . . . .	2089
6.695.3 Member Enumeration Documentation . . . . .	2089
6.695.3.1 anonymous enum . . . . .	2089
6.696 Rcpp::Rstreambuf< OUTPUT > Class Template Reference . . . . .	2090
6.696.1 Detailed Description . . . . .	2091
6.696.2 Constructor & Destructor Documentation . . . . .	2091
6.696.2.1 Rstreambuf() . . . . .	2091
6.696.3 Member Function Documentation . . . . .	2091
6.696.3.1 overflow() [1/3] . . . . .	2091
6.696.3.2 overflow() [2/3] . . . . .	2092
6.696.3.3 overflow() [3/3] . . . . .	2092
6.696.3.4 sync() [1/3] . . . . .	2092
6.696.3.5 sync() [2/3] . . . . .	2092
6.696.3.6 sync() [3/3] . . . . .	2092
6.696.3.7 xspuhn() [1/3] . . . . .	2092
6.696.3.8 xspuhn() [2/3] . . . . .	2093

---

6.696.3.9 xspuhn() [3/3]	2093
6.697 Rcpp::algorithm::helpers::rtype< T > Struct Template Reference	2093
6.697.1 Detailed Description	2094
6.697.2 Member Typedef Documentation	2094
6.697.2.1 helper_type	2094
6.697.2.2 type	2094
6.697.3 Member Function Documentation	2094
6.697.3.1 NA()	2094
6.697.3.2 ONE()	2095
6.697.3.3 ZERO()	2095
6.697.4 Member Data Documentation	2095
6.697.4.1 RTYPE	2095
6.698 Rcpp::algorithm::helpers::rtype_helper< T > Struct Template Reference	2095
6.698.1 Detailed Description	2095
6.699 Rcpp::algorithm::helpers::rtype_helper< double > Struct Reference	2096
6.699.1 Detailed Description	2096
6.699.2 Member Typedef Documentation	2096
6.699.2.1 type	2096
6.699.3 Member Function Documentation	2096
6.699.3.1 NA()	2096
6.699.3.2 ONE()	2097
6.699.3.3 ZERO()	2097
6.699.4 Member Data Documentation	2097
6.699.4.1 RTYPE	2097
6.700 Rcpp::algorithm::helpers::rtype_helper< int > Struct Reference	2097
6.700.1 Detailed Description	2098
6.700.2 Member Typedef Documentation	2098
6.700.2.1 type	2098
6.700.3 Member Function Documentation	2098
6.700.3.1 NA()	2098
6.700.3.2 ONE()	2098
6.700.3.3 ZERO()	2099
6.700.4 Member Data Documentation	2099
6.700.4.1 RTYPE	2099
6.701 Rcpp::rule Struct Reference	2099
6.701.1 Detailed Description	2099
6.701.2 Member Data Documentation	2099
6.701.2.1 r_day	2100
6.701.2.2 r_mon	2100

---

6.701.2.3 r_time . . . . .	2100
6.701.2.4 r_type . . . . .	2100
6.701.2.5 r_week . . . . .	2100
6.702 Rcpp::S4_CppConstructor< Class > Class Template Reference . . . . .	2101
6.702.1 Detailed Description . . . . .	2102
6.702.2 Member Typedef Documentation . . . . .	2102
6.702.2.1 Base . . . . .	2102
6.702.2.2 Storage . . . . .	2102
6.702.2.3 XP_Class . . . . .	2102
6.702.3 Constructor & Destructor Documentation . . . . .	2102
6.702.3.1 S4_CppConstructor() . . . . .	2103
6.703 Rcpp::S4_CppOverloadedMethods< Class > Class Template Reference . . . . .	2103
6.703.1 Detailed Description . . . . .	2105
6.703.2 Member Typedef Documentation . . . . .	2105
6.703.2.1 Base . . . . .	2105
6.703.2.2 signed_method_class . . . . .	2105
6.703.2.3 vec_signed_method . . . . .	2105
6.703.2.4 XP_Class . . . . .	2105
6.703.3 Constructor & Destructor Documentation . . . . .	2106
6.703.3.1 S4_CppOverloadedMethods() . . . . .	2106
6.704 Rcpp::S4_field< Class > Class Template Reference . . . . .	2107
6.704.1 Detailed Description . . . . .	2108
6.704.2 Member Typedef Documentation . . . . .	2108
6.704.2.1 Base . . . . .	2108
6.704.2.2 XP_Class . . . . .	2108
6.704.3 Constructor & Destructor Documentation . . . . .	2108
6.704.3.1 S4_field() . . . . .	2109
6.705 Rcpp::traits::same_type< T, U > Struct Template Reference . . . . .	2109
6.705.1 Detailed Description . . . . .	2110
6.706 Rcpp::traits::same_type< T, T > Struct Template Reference . . . . .	2111
6.706.1 Detailed Description . . . . .	2111
6.707 Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION > Class Template Reference . . . . .	2112
6.707.1 Detailed Description . . . . .	2113
6.707.2 Member Typedef Documentation . . . . .	2113
6.707.2.1 converter_type . . . . .	2113
6.707.2.2 EXT . . . . .	2113
6.707.2.3 result_type . . . . .	2113
6.707.2.4 STORAGE . . . . .	2114
6.707.2.5 VEC . . . . .	2114

---

6.707.3 Constructor & Destructor Documentation	2114
6.707.3.1 Sapply()	2114
6.707.4 Member Function Documentation	2114
6.707.4.1 operator[]()	2115
6.707.4.2 size()	2115
6.707.5 Member Data Documentation	2115
6.707.5.1 fun	2115
6.707.5.2 RESULT_R_TYPE	2116
6.707.5.3 vec	2116
6.708 Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true > Class Template Reference	2116
6.708.1 Detailed Description	2117
6.708.2 Member Typedef Documentation	2117
6.708.2.1 EXT	2117
6.708.2.2 result_type	2118
6.708.2.3 STORAGE	2118
6.708.2.4 VEC	2118
6.708.3 Constructor & Destructor Documentation	2118
6.708.3.1 Sapply()	2118
6.708.4 Member Function Documentation	2118
6.708.4.1 operator[]()	2119
6.708.4.2 size()	2119
6.708.5 Member Data Documentation	2119
6.708.5.1 fun	2119
6.708.5.2 RESULT_R_TYPE	2119
6.708.5.3 vec	2120
6.709 Rcpp::sugar::sapply_application_result_of< Function, SugarExpression > Struct Template Reference	2120
6.709.1 Detailed Description	2120
6.709.2 Member Typedef Documentation	2120
6.709.2.1 type	2120
6.710 Rcpp::sugar::cbind_impl::scalar< RTYPE > Struct Template Reference	2121
6.710.1 Detailed Description	2121
6.710.2 Member Typedef Documentation	2121
6.710.2.1 type	2121
6.711 Rcpp::sugar::cbind_impl::ScalarBindable< T > Class Template Reference	2122
6.711.1 Detailed Description	2123
6.711.2 Member Typedef Documentation	2123
6.711.2.1 stored_type	2123
6.711.3 Member Enumeration Documentation	2123
6.711.3.1 anonymous enum	2123



---

6.711.4 Constructor & Destructor Documentation	2123
6.711.4.1 ScalarBindable()	2123
6.711.5 Member Function Documentation	2124
6.711.5.1 ncol()	2124
6.711.5.2 nrow()	2124
6.711.5.3 operator>()	2124
6.711.5.4 operator[]()	2124
6.711.5.5 size()	2125
6.711.6 Member Data Documentation	2125
6.711.6.1 t	2125
6.712 Rcpp::sugar::Sd< RTYPE, NA, T > Class Template Reference	2125
6.712.1 Detailed Description	2126
6.712.2 Member Typedef Documentation	2126
6.712.2.1 STORAGE	2126
6.712.2.2 VEC_TYPE	2127
6.712.3 Constructor & Destructor Documentation	2127
6.712.3.1 Sd()	2127
6.712.4 Member Function Documentation	2127
6.712.4.1 get()	2127
6.712.5 Member Data Documentation	2128
6.712.5.1 object	2128
6.713 Rcpp::sugar::SelfHash< RTYPE > Class Template Reference	2128
6.713.1 Detailed Description	2129
6.713.2 Member Typedef Documentation	2129
6.713.2.1 STORAGE	2129
6.713.2.2 VECTOR	2129
6.713.3 Constructor & Destructor Documentation	2129
6.713.3.1 SelfHash()	2129
6.713.4 Member Function Documentation	2130
6.713.4.1 add_value_get_index()	2130
6.713.4.2 fill_and_self_match()	2130
6.713.4.3 get_addr() [1/4]	2131
6.713.4.4 get_addr() [2/4]	2131
6.713.4.5 get_addr() [3/4]	2131
6.713.4.6 get_addr() [4/4]	2132
6.713.4.7 get_index()	2132
6.713.4.8 size()	2132
6.713.5 Member Data Documentation	2132
6.713.5.1 data	2133

---

6.713.5.2 indices	2133
6.713.5.3 k	2133
6.713.5.4 m	2133
6.713.5.5 n	2134
6.713.5.6 size_	2134
6.713.5.7 src	2134
6.714 Rcpp::sugar::SelfInserter< HASH, STORAGE > Class Template Reference	2134
6.714.1 Detailed Description	2135
6.714.2 Constructor & Destructor Documentation	2135
6.714.2.1 SelfInserter()	2135
6.714.3 Member Function Documentation	2135
6.714.3.1 operator>()	2135
6.714.4 Member Data Documentation	2136
6.714.4.1 hash	2136
6.714.4.2 index	2136
6.715 Rcpp::sugar::SelfMatch< RTYPE, TABLE_T > Class Template Reference	2136
6.715.1 Detailed Description	2137
6.715.2 Member Typedef Documentation	2137
6.715.2.1 HASH	2137
6.715.2.2 Inserter	2138
6.715.2.3 STORAGE	2138
6.715.3 Constructor & Destructor Documentation	2138
6.715.3.1 SelfMatch()	2138
6.715.4 Member Function Documentation	2139
6.715.4.1 operator IntegerVector()	2139
6.715.5 Member Data Documentation	2139
6.715.5.1 hash	2139
6.715.5.2 result	2139
6.716 Rcpp::sugar::SeqLen Class Reference	2140
6.716.1 Detailed Description	2141
6.716.2 Constructor & Destructor Documentation	2141
6.716.2.1 SeqLen()	2141
6.716.3 Member Function Documentation	2141
6.716.3.1 operator[]()	2141
6.716.3.2 size()	2142
6.716.4 Member Data Documentation	2142
6.716.4.1 len	2142
6.717 Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	2142
6.717.1 Detailed Description	2143

---

6.717.2 Member Typedef Documentation	2143
6.717.2.1 ITERATOR	2143
6.717.2.2 SET	2143
6.717.2.3 STORAGE	2143
6.717.3 Constructor & Destructor Documentation	2144
6.717.3.1 SetDiff()	2144
6.717.4 Member Function Documentation	2144
6.717.4.1 get()	2144
6.717.5 Member Data Documentation	2145
6.717.5.1 lhs_set	2145
6.717.5.2 rhs_set	2145
6.718 Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	2145
6.718.1 Detailed Description	2146
6.718.2 Member Typedef Documentation	2146
6.718.2.1 ITERATOR	2146
6.718.2.2 SET	2146
6.718.2.3 STORAGE	2147
6.718.3 Constructor & Destructor Documentation	2147
6.718.3.1 SetEqual()	2147
6.718.4 Member Function Documentation	2147
6.718.4.1 get()	2147
6.718.5 Member Data Documentation	2147
6.718.5.1 lhs_set	2148
6.718.5.2 rhs_set	2148
6.719 Rcpp::Shelter< T > Class Template Reference	2148
6.719.1 Detailed Description	2149
6.719.2 Constructor & Destructor Documentation	2149
6.719.2.1 Shelter() [1/2]	2149
6.719.2.2 ~Shelter()	2149
6.719.2.3 Shelter() [2/2]	2150
6.719.3 Member Function Documentation	2150
6.719.3.1 operator>()	2150
6.719.3.2 operator=()	2150
6.719.4 Member Data Documentation	2150
6.719.4.1 nprotected	2151
6.720 Rcpp::Shield< T > Class Template Reference	2151
6.720.1 Detailed Description	2151
6.720.2 Constructor & Destructor Documentation	2151
6.720.2.1 Shield() [1/2]	2152

---

6.720.2.2 ~Shield()	2152
6.720.2.3 Shield() [2/2]	2152
6.720.3 Member Function Documentation	2152
6.720.3.1 operator SEXP()	2153
6.720.3.2 operator=()	2153
6.720.4 Member Data Documentation	2153
6.720.4.1 t	2153
6.721 Rcpp::sugar::Sign< RTYPE, NA, T > Class Template Reference	2154
6.721.1 Detailed Description	2155
6.721.2 Member Typedef Documentation	2155
6.721.2.1 r_import_type	2155
6.721.2.2 STORAGE	2155
6.721.2.3 VEC_TYPE	2155
6.721.3 Constructor & Destructor Documentation	2155
6.721.3.1 Sign()	2156
6.721.4 Member Function Documentation	2156
6.721.4.1 get()	2156
6.721.4.2 operator SEXP()	2157
6.721.4.3 operator[]()	2157
6.721.4.4 size()	2158
6.721.5 Member Data Documentation	2158
6.721.5.1 object	2158
6.722 Rcpp::sugar::sign__impl< NA, RTYPE > Class Template Reference	2158
6.722.1 Detailed Description	2158
6.722.2 Member Typedef Documentation	2159
6.722.2.1 STORAGE	2159
6.722.3 Member Function Documentation	2159
6.722.3.1 get()	2159
6.723 Rcpp::sugar::sign__impl< false, RTYPE > Class Template Reference	2159
6.723.1 Detailed Description	2160
6.723.2 Member Typedef Documentation	2160
6.723.2.1 STORAGE	2160
6.723.3 Member Function Documentation	2160
6.723.3.1 get()	2160
6.724 Rcpp::SignedConstructor< Class > Class Template Reference	2160
6.724.1 Detailed Description	2161
6.724.2 Constructor & Destructor Documentation	2161
6.724.2.1 SignedConstructor()	2161
6.724.3 Member Function Documentation	2161

---

6.724.3.1 nargs()	2161
6.724.3.2 signature()	2162
6.724.4 Member Data Documentation	2162
6.724.4.1 ctor	2162
6.724.4.2 docstring	2162
6.724.4.3 valid	2162
6.725 Rcpp::SignedFactory< Class > Class Template Reference	2163
6.725.1 Detailed Description	2163
6.725.2 Constructor & Destructor Documentation	2163
6.725.2.1 SignedFactory()	2163
6.725.3 Member Function Documentation	2163
6.725.3.1 nargs()	2164
6.725.3.2 signature()	2164
6.725.4 Member Data Documentation	2164
6.725.4.1 docstring	2164
6.725.4.2 fact	2164
6.725.4.3 valid	2165
6.726 Rcpp::SignedMethod< Class > Class Template Reference	2165
6.726.1 Detailed Description	2166
6.726.2 Member Typedef Documentation	2166
6.726.2.1 METHOD	2166
6.726.3 Constructor & Destructor Documentation	2166
6.726.3.1 SignedMethod()	2166
6.726.4 Member Function Documentation	2166
6.726.4.1 is_const()	2167
6.726.4.2 is_void()	2167
6.726.4.3 nargs()	2168
6.726.4.4 signature()	2168
6.726.5 Member Data Documentation	2169
6.726.5.1 docstring	2169
6.726.5.2 method	2169
6.726.5.3 valid	2169
6.727 Rcpp::stats::SignRankGenerator Class Reference	2170
6.727.1 Detailed Description	2171
6.727.2 Constructor & Destructor Documentation	2171
6.727.2.1 SignRankGenerator()	2171
6.727.3 Member Function Documentation	2171
6.727.3.1 operator>()	2171
6.727.4 Member Data Documentation	2171

---

6.727.4.1 nn	2171
6.728 Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy > Class Template Reference	2172
6.728.1 Detailed Description	2173
6.728.2 Member Typedef Documentation	2173
6.728.2.1 CTYPE	2173
6.728.2.2 VECTOR	2173
6.728.3 Constructor & Destructor Documentation	2173
6.728.3.1 simple_name_proxy() [1/2]	2174
6.728.3.2 simple_name_proxy() [2/2]	2174
6.728.3.3 ~simple_name_proxy()	2174
6.728.4 Member Function Documentation	2174
6.728.4.1 get()	2174
6.728.4.2 operator CTYPE()	2175
6.728.4.3 operator SEXP()	2175
6.728.4.4 operator=() [1/3]	2176
6.728.4.5 operator=() [2/3]	2176
6.728.4.6 operator=() [3/3]	2177
6.728.4.7 set()	2177
6.728.5 Member Data Documentation	2178
6.728.5.1 name	2178
6.728.5.2 parent	2178
6.729 Rcpp::SingleLogicalResult< NA, T > Class Template Reference	2178
6.729.1 Detailed Description	2178
6.730 Rcpp::sugar::SingleLogicalResult< NA, T > Class Template Reference	2179
6.730.1 Detailed Description	2179
6.730.2 Constructor & Destructor Documentation	2180
6.730.2.1 SingleLogicalResult()	2180
6.730.3 Member Function Documentation	2180
6.730.3.1 apply()	2180
6.730.3.2 get()	2180
6.730.3.3 get_sexp()	2181
6.730.3.4 is_false()	2182
6.730.3.5 is_na()	2182
6.730.3.6 is_true()	2183
6.730.3.7 is_unresolved()	2183
6.730.3.8 operator bool()	2183
6.730.3.9 operator SEXP()	2184
6.730.3.10 reset()	2184
6.730.3.11 set()	2185

6.730.3.12 set_false()	2185
6.730.3.13 set_na()	2186
6.730.3.14 set_true()	2186
6.730.3.15 size()	2187
6.730.4 Member Data Documentation	2187
6.730.4.1 result	2187
6.730.4.2 UNRESOLVED	2187
6.731 Rcpp::SlotProxyPolicy< CLASS >::SlotProxy Class Reference	2188
6.731.1 Detailed Description	2189
6.731.2 Constructor & Destructor Documentation	2189
6.731.2.1 SlotProxy()	2189
6.731.3 Member Function Documentation	2189
6.731.3.1 get()	2189
6.731.3.2 operator SEXP()	2190
6.731.3.3 operator T()	2190
6.731.3.4 operator=() [1/3]	2191
6.731.3.5 operator=() [2/3]	2191
6.731.3.6 operator=() [3/3]	2192
6.731.3.7 set()	2192
6.731.4 Member Data Documentation	2192
6.731.4.1 parent	2192
6.731.4.2 slot_name	2192
6.732 Rcpp::SlotProxyPolicy< CLASS > Class Template Reference	2193
6.732.1 Detailed Description	2193
6.732.2 Member Function Documentation	2194
6.732.2.1 hasSlot()	2194
6.732.2.2 slot() [1/2]	2194
6.732.2.3 slot() [2/2]	2194
6.733 Rcpp::attributes::SourceFileAttributes Class Reference	2194
6.733.1 Detailed Description	2195
6.733.2 Member Typedef Documentation	2195
6.733.2.1 const_iterator	2195
6.733.3 Constructor & Destructor Documentation	2195
6.733.3.1 ~SourceFileAttributes()	2195
6.733.4 Member Function Documentation	2196
6.733.4.1 begin()	2196
6.733.4.2 end()	2196
6.733.4.3 hasGeneratorOutput()	2196
6.733.4.4 hasInterface()	2196

---

6.733.4.5	hasPackageInit()	2197
6.733.4.6	modules()	2197
6.733.4.7	roxygenChunks()	2197
6.733.4.8	sourceFile()	2197
6.734	Rcpp::attributes::SourceFileAttributesParser Class Reference	2198
6.734.1	Detailed Description	2199
6.734.2	Constructor & Destructor Documentation	2199
6.734.2.1	SourceFileAttributesParser() [1/2]	2199
6.734.2.2	SourceFileAttributesParser() [2/2]	2200
6.734.3	Member Function Documentation	2200
6.734.3.1	attributeWarning() [1/2]	2200
6.734.3.2	attributeWarning() [2/2]	2200
6.734.3.3	begin()	2200
6.734.3.4	embeddedR()	2201
6.734.3.5	end()	2201
6.734.3.6	hasGeneratorOutput()	2201
6.734.3.7	hasInterface()	2202
6.734.3.8	hasPackageInit()	2202
6.734.3.9	isKnownAttribute()	2202
6.734.3.10	modules()	2203
6.734.3.11	operator=()	2203
6.734.3.12	parseArguments()	2203
6.734.3.13	parseAttribute()	2203
6.734.3.14	parseFunction()	2203
6.734.3.15	parseParameters()	2204
6.734.3.16	parseSignature()	2204
6.734.3.17	parseType()	2204
6.734.3.18	rcppExportInvalidParameterWarning()	2204
6.734.3.19	rcppExportNoFunctionFoundWarning()	2204
6.734.3.20	rcppExportWarning()	2204
6.734.3.21	rcppInterfacesWarning()	2205
6.734.3.22	roxygenChunks()	2205
6.734.3.23	sourceDependencies()	2205
6.734.3.24	sourceFile()	2205
6.734.4	Member Data Documentation	2205
6.734.4.1	attributes_	2206
6.734.4.2	embeddedR_	2206
6.734.4.3	hasPackageInit_	2206
6.734.4.4	lines_	2206



---

6.734.4.5 modules_ . . . . .	2206
6.734.4.6 roxygenBuffer_ . . . . .	2207
6.734.4.7 roxygenChunks_ . . . . .	2207
6.734.4.8 sourceDependencies_ . . . . .	2207
6.734.4.9 sourceFile_ . . . . .	2207
6.735 Rcpp::algorithm::helpers::sqrt Struct Reference . . . . .	2207
6.735.1 Detailed Description . . . . .	2208
6.735.2 Member Function Documentation . . . . .	2208
6.735.2.1 operator>() . . . . .	2208
6.736 Rcpp::state Struct Reference . . . . .	2209
6.736.1 Detailed Description . . . . .	2209
6.736.2 Member Data Documentation . . . . .	2209
6.736.2.1 ats . . . . .	2209
6.736.2.2 charcnt . . . . .	2210
6.736.2.3 chars . . . . .	2210
6.736.2.4 goahead . . . . .	2210
6.736.2.5 goback . . . . .	2210
6.736.2.6 leapcnt . . . . .	2210
6.736.2.7 lsis . . . . .	2211
6.736.2.8 timecnt . . . . .	2211
6.736.2.9 ttis . . . . .	2211
6.736.2.10 typecnt . . . . .	2211
6.736.2.11 types . . . . .	2211
6.737 Rcpp::traits::storage_type< RTYPE > Struct Template Reference . . . . .	2212
6.737.1 Detailed Description . . . . .	2212
6.737.2 Member Typedef Documentation . . . . .	2212
6.737.2.1 type . . . . .	2213
6.738 Rcpp::traits::storage_type< CPLXSP > Struct Reference . . . . .	2213
6.738.1 Detailed Description . . . . .	2213
6.738.2 Member Typedef Documentation . . . . .	2213
6.738.2.1 type . . . . .	2213
6.739 Rcpp::traits::storage_type< INTSP > Struct Reference . . . . .	2213
6.739.1 Detailed Description . . . . .	2214
6.739.2 Member Typedef Documentation . . . . .	2214
6.739.2.1 type . . . . .	2214
6.740 Rcpp::traits::storage_type< LGLSP > Struct Reference . . . . .	2214
6.740.1 Detailed Description . . . . .	2214
6.740.2 Member Typedef Documentation . . . . .	2214
6.740.2.1 type . . . . .	2215

---

6.741 Rcpp::traits::storage_type< RAWSXP > Struct Reference	2215
6.741.1 Detailed Description	2215
6.741.2 Member Typedef Documentation	2215
6.741.2.1 type	2215
6.742 Rcpp::traits::storage_type< REALSXP > Struct Reference	2215
6.742.1 Detailed Description	2216
6.742.2 Member Typedef Documentation	2216
6.742.2.1 type	2216
6.743 Rcpp::String Class Reference	2216
6.743.1 Detailed Description	2219
6.743.2 Member Typedef Documentation	2219
6.743.2.1 const_StringProxy	2219
6.743.2.2 StringProxy	2219
6.743.3 Constructor & Destructor Documentation	2219
6.743.3.1 String() [1/16]	2219
6.743.3.2 String() [2/16]	2220
6.743.3.3 String() [3/16]	2220
6.743.3.4 String() [4/16]	2221
6.743.3.5 String() [5/16]	2221
6.743.3.6 String() [6/16]	2222
6.743.3.7 String() [7/16]	2222
6.743.3.8 String() [8/16]	2223
6.743.3.9 String() [9/16]	2223
6.743.3.10 String() [10/16]	2223
6.743.3.11 String() [11/16]	2224
6.743.3.12 String() [12/16]	2224
6.743.3.13 String() [13/16]	2225
6.743.3.14 String() [14/16]	2225
6.743.3.15 String() [15/16]	2226
6.743.3.16 String() [16/16]	2226
6.743.3.17 ~String()	2227
6.743.4 Member Function Documentation	2227
6.743.4.1 append()	2227
6.743.4.2 append_wide_string()	2227
6.743.4.3 assign_wide_string()	2228
6.743.4.4 get_cstring()	2228
6.743.4.5 get_encoding()	2229
6.743.4.6 get_sexp() [1/2]	2229
6.743.4.7 get_sexp() [2/2]	2229

---

6.743.4.8 <code>get_sexp_impl()</code> . . . . .	2230
6.743.4.9 <code>is_na()</code> . . . . .	2230
6.743.4.10 <code>operator std::string()</code> . . . . .	2230
6.743.4.11 <code>operator std::wstring()</code> . . . . .	2231
6.743.4.12 <code>operator!=( )</code> [1/4] . . . . .	2231
6.743.4.13 <code>operator!=( )</code> [2/4] . . . . .	2232
6.743.4.14 <code>operator!=( )</code> [3/4] . . . . .	2232
6.743.4.15 <code>operator!=( )</code> [4/4] . . . . .	2233
6.743.4.16 <code>operator+=( )</code> [1/8] . . . . .	2233
6.743.4.17 <code>operator+=( )</code> [2/8] . . . . .	2234
6.743.4.18 <code>operator+=( )</code> [3/8] . . . . .	2234
6.743.4.19 <code>operator+=( )</code> [4/8] . . . . .	2235
6.743.4.20 <code>operator+=( )</code> [5/8] . . . . .	2235
6.743.4.21 <code>operator+=( )</code> [6/8] . . . . .	2236
6.743.4.22 <code>operator+=( )</code> [7/8] . . . . .	2236
6.743.4.23 <code>operator+=( )</code> [8/8] . . . . .	2237
6.743.4.24 <code>operator&lt;()</code> . . . . .	2237
6.743.4.25 <code>operator=( )</code> [1/12] . . . . .	2238
6.743.4.26 <code>operator=( )</code> [2/12] . . . . .	2238
6.743.4.27 <code>operator=( )</code> [3/12] . . . . .	2238
6.743.4.28 <code>operator=( )</code> [4/12] . . . . .	2239
6.743.4.29 <code>operator=( )</code> [5/12] . . . . .	2239
6.743.4.30 <code>operator=( )</code> [6/12] . . . . .	2240
6.743.4.31 <code>operator=( )</code> [7/12] . . . . .	2240
6.743.4.32 <code>operator=( )</code> [8/12] . . . . .	2241
6.743.4.33 <code>operator=( )</code> [9/12] . . . . .	2241
6.743.4.34 <code>operator=( )</code> [10/12] . . . . .	2242
6.743.4.35 <code>operator=( )</code> [11/12] . . . . .	2242
6.743.4.36 <code>operator=( )</code> [12/12] . . . . .	2243
6.743.4.37 <code>operator==( )</code> [1/4] . . . . .	2243
6.743.4.38 <code>operator==( )</code> [2/4] . . . . .	2244
6.743.4.39 <code>operator==( )</code> [3/4] . . . . .	2244
6.743.4.40 <code>operator==( )</code> [4/4] . . . . .	2245
6.743.4.41 <code>operator&gt;()</code> . . . . .	2245
6.743.4.42 <code>push_back()</code> [1/3] . . . . .	2246
6.743.4.43 <code>push_back()</code> [2/3] . . . . .	2246
6.743.4.44 <code>push_back()</code> [3/3] . . . . .	2247
6.743.4.45 <code>push_front()</code> [1/3] . . . . .	2247
6.743.4.46 <code>push_front()</code> [2/3] . . . . .	2248

---

6.743.4.47 push_front() [3/3]	2248
6.743.4.48 replace_all() [1/5]	2249
6.743.4.49 replace_all() [2/5]	2249
6.743.4.50 replace_all() [3/5]	2250
6.743.4.51 replace_all() [4/5]	2250
6.743.4.52 replace_all() [5/5]	2251
6.743.4.53 replace_first() [1/4]	2251
6.743.4.54 replace_first() [2/4]	2252
6.743.4.55 replace_first() [3/4]	2252
6.743.4.56 replace_first() [4/4]	2253
6.743.4.57 replace_last() [1/4]	2253
6.743.4.58 replace_last() [2/4]	2254
6.743.4.59 replace_last() [3/4]	2254
6.743.4.60 replace_last() [4/4]	2255
6.743.4.61 set_encoding()	2255
6.743.4.62 set_na()	2256
6.743.4.63 setBuffer()	2256
6.743.4.64 setData()	2257
6.743.5 Member Data Documentation	2257
6.743.5.1 buffer	2257
6.743.5.2 buffer_ready	2257
6.743.5.3 data	2258
6.743.5.4 enc	2258
6.743.5.5 token	2258
6.743.5.6 valid	2258
6.744 Rcpp::internal::string_element_converter< RTYPE > Class Template Reference	2259
6.744.1 Detailed Description	2259
6.744.2 Member Typedef Documentation	2259
6.744.2.1 target	2259
6.744.3 Member Function Documentation	2259
6.744.3.1 get() [1/5]	2260
6.744.3.2 get() [2/5]	2260
6.744.3.3 get() [3/5]	2260
6.744.3.4 get() [4/5]	2261
6.744.3.5 get() [5/5]	2261
6.745 Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy > Class Template Reference	2261
6.745.1 Detailed Description	2262
6.745.2 Member Typedef Documentation	2262
6.745.2.1 iterator	2263

---

6.745.2.2 reference	2263
6.745.2.3 VECTOR	2263
6.745.3 Constructor & Destructor Documentation	2263
6.745.3.1 string_name_proxy() [1/2]	2263
6.745.3.2 string_name_proxy() [2/2]	2264
6.745.3.3 ~string_name_proxy()	2264
6.745.4 Member Function Documentation	2264
6.745.4.1 begin()	2264
6.745.4.2 end()	2265
6.745.4.3 get()	2265
6.745.4.4 operator char *()	2266
6.745.4.5 operator SEXP()	2266
6.745.4.6 operator=() [1/2]	2267
6.745.4.7 operator=() [2/2]	2267
6.745.4.8 operator[]()	2268
6.745.4.9 set()	2268
6.745.4.10 size()	2269
6.745.5 Member Data Documentation	2269
6.745.5.1 name	2269
6.745.5.2 parent	2269
6.746 Rcpp::internal::string_proxy< RTYPE, StoragePolicy > Class Template Reference	2270
6.746.1 Detailed Description	2272
6.746.2 Member Typedef Documentation	2272
6.746.2.1 iterator	2272
6.746.2.2 reference	2272
6.746.2.3 VECTOR	2272
6.746.3 Constructor & Destructor Documentation	2272
6.746.3.1 string_proxy() [1/3]	2273
6.746.3.2 string_proxy() [2/3]	2273
6.746.3.3 string_proxy() [3/3]	2273
6.746.4 Member Function Documentation	2273
6.746.4.1 apply() [1/2]	2274
6.746.4.2 apply() [2/2]	2274
6.746.4.3 begin()	2275
6.746.4.4 empty()	2275
6.746.4.5 end()	2276
6.746.4.6 get()	2276
6.746.4.7 import()	2276
6.746.4.8 move()	2277

---

6.746.4.9 operator char *()	2277
6.746.4.10 operator SEXP()	2277
6.746.4.11 operator"!=( ) [1/3]	2278
6.746.4.12 operator"!=( ) [2/3]	2278
6.746.4.13 operator"!=( ) [3/3]	2279
6.746.4.14 operator+=( ) [1/2]	2280
6.746.4.15 operator+=( ) [2/2]	2280
6.746.4.16 operator=( ) [1/9]	2280
6.746.4.17 operator=( ) [2/9]	2281
6.746.4.18 operator=( ) [3/9]	2281
6.746.4.19 operator=( ) [4/9]	2282
6.746.4.20 operator=( ) [5/9]	2282
6.746.4.21 operator=( ) [6/9]	2283
6.746.4.22 operator=( ) [7/9]	2283
6.746.4.23 operator=( ) [8/9]	2284
6.746.4.24 operator=( ) [9/9]	2285
6.746.4.25 operator==( ) [1/3]	2285
6.746.4.26 operator==( ) [2/3]	2286
6.746.4.27 operator==( ) [3/3]	2286
6.746.4.28 operator[]()	2287
6.746.4.29 set() [1/2]	2287
6.746.4.30 set() [2/2]	2287
6.746.4.31 size()	2288
6.746.4.32 swap()	2288
6.746.4.33 transform()	2289
6.746.5 Friends And Related Function Documentation	2289
6.746.5.1 operator+	2289
6.746.5.2 operator<<	2289
6.746.6 Member Data Documentation	2290
6.746.6.1 buffer	2290
6.746.6.2 index	2290
6.746.6.3 parent	2290
6.747 Rcpp::StringTransformer< UnaryOperator > Class Template Reference	2291
6.747.1 Detailed Description	2292
6.747.2 Constructor & Destructor Documentation	2292
6.747.2.1 StringTransformer()	2292
6.747.2.2 ~StringTransformer()	2292
6.747.3 Member Function Documentation	2292
6.747.3.1 operator>()()	2292

---

6.747.4 Member Data Documentation	2293
6.747.4.1 buffer	2293
6.747.4.2 op	2293
6.748 Rcpp::SubMatrix< RTYPE > Class Template Reference	2293
6.748.1 Detailed Description	2294
6.748.2 Member Typedef Documentation	2295
6.748.2.1 MATRIX	2295
6.748.2.2 Proxy	2295
6.748.2.3 vec_iterator	2295
6.748.3 Constructor & Destructor Documentation	2295
6.748.3.1 SubMatrix()	2295
6.748.4 Member Function Documentation	2296
6.748.4.1 column_iterator()	2296
6.748.4.2 ncol()	2296
6.748.4.3 nrow()	2296
6.748.4.4 operator>()()	2297
6.748.4.5 size()	2297
6.748.5 Member Data Documentation	2297
6.748.5.1 iter	2297
6.748.5.2 m	2298
6.748.5.3 m_nr	2298
6.748.5.4 nc	2298
6.748.5.5 nr	2298
6.749 Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T > Class Template Reference	2299
6.749.1 Detailed Description	2300
6.749.2 Member Typedef Documentation	2300
6.749.2.1 LHS_t	2300
6.749.2.2 RHS_t	2301
6.749.3 Constructor & Destructor Documentation	2301
6.749.3.1 SubsetProxy() [1/2]	2301
6.749.3.2 SubsetProxy() [2/2]	2301
6.749.4 Member Function Documentation	2302
6.749.4.1 check_indices()	2302
6.749.4.2 get_indices() [1/4]	2302
6.749.4.3 get_indices() [2/4]	2303
6.749.4.4 get_indices() [3/4]	2303
6.749.4.5 get_indices() [4/4]	2304
6.749.4.6 get_vec()	2305

6.749.4.7 operator SEXP()	2305
6.749.4.8 operator Vector< RTYPE, StoragePolicy >()	2306
6.749.4.9 operator=() [1/7]	2307
6.749.4.10 operator=() [2/7]	2307
6.749.4.11 operator=() [3/7]	2307
6.749.4.12 operator=() [4/7]	2308
6.749.4.13 operator=() [5/7]	2308
6.749.4.14 operator=() [6/7]	2308
6.749.4.15 operator=() [7/7]	2309
6.749.5 Member Data Documentation	2309
6.749.5.1 indices	2309
6.749.5.2 indices_n	2309
6.749.5.3 lhs	2310
6.749.5.4 lhs_n	2310
6.749.5.5 rhs	2310
6.749.5.6 rhs_n	2310
6.750 tinyformat::detail::is_convertible< T1, T2 >::succeed Struct Reference	2311
6.750.1 Detailed Description	2311
6.750.2 Member Data Documentation	2311
6.750.2.1 dummy	2311
6.751 Rcpp::sugar::sugar_const_iterator_type< T > Struct Template Reference	2311
6.751.1 Detailed Description	2312
6.751.2 Member Typedef Documentation	2312
6.751.2.1 type	2312
6.752 Rcpp::sugar::sugar_const_iterator_type< CharacterVector > Struct Reference	2312
6.752.1 Detailed Description	2312
6.752.2 Member Typedef Documentation	2312
6.752.2.1 type	2313
6.753 Rcpp::sugar::sugar_const_iterator_type< Rcpp::Vector< RTYPE > > Struct Template Reference	2313
6.753.1 Detailed Description	2313
6.753.2 Member Typedef Documentation	2313
6.753.2.1 type	2313
6.754 Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 > Class Template Reference	2314
6.754.1 Detailed Description	2314
6.754.2 Member Typedef Documentation	2315
6.754.2.1 FunPtr	2315
6.754.3 Constructor & Destructor Documentation	2315
6.754.3.1 SugarBlock_1()	2315
6.754.4 Member Function Documentation	2315



---

6.754.4.1 operator[]()	2315
6.754.4.2 size()	2316
6.754.5 Member Data Documentation	2316
6.754.5.1 ptr	2316
6.754.5.2 vec	2316
6.755 Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 > Class Template Reference	2317
6.755.1 Detailed Description	2317
6.755.2 Member Typedef Documentation	2318
6.755.2.1 FunPtr	2318
6.755.3 Constructor & Destructor Documentation	2318
6.755.3.1 SugarBlock_2()	2318
6.755.4 Member Function Documentation	2318
6.755.4.1 operator[]()	2318
6.755.4.2 size()	2319
6.755.5 Member Data Documentation	2319
6.755.5.1 ptr	2319
6.755.5.2 x	2319
6.755.5.3 y	2319
6.756 Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 > Class Template Reference	2320
6.756.1 Detailed Description	2320
6.756.2 Member Typedef Documentation	2321
6.756.2.1 FunPtr	2321
6.756.3 Constructor & Destructor Documentation	2321
6.756.3.1 SugarBlock_2__PV()	2321
6.756.4 Member Function Documentation	2321
6.756.4.1 operator[]()	2321
6.756.4.2 size()	2322
6.756.5 Member Data Documentation	2322
6.756.5.1 ptr	2322
6.756.5.2 x	2322
6.756.5.3 y	2322
6.757 Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 > Class Template Reference	2323
6.757.1 Detailed Description	2323
6.757.2 Member Typedef Documentation	2324
6.757.2.1 FunPtr	2324
6.757.3 Constructor & Destructor Documentation	2324
6.757.3.1 SugarBlock_2__VP()	2324
6.757.4 Member Function Documentation	2324
6.757.4.1 operator[]()	2324

---

6.757.4.2 size()	2325
6.757.5 Member Data Documentation	2325
6.757.5.1 ptr	2325
6.757.5.2 x	2325
6.757.5.3 y	2325
6.758 Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 > Class Template Reference	2326
6.758.1 Detailed Description	2327
6.758.2 Member Typedef Documentation	2327
6.758.2.1 FunPtr	2327
6.758.3 Constructor & Destructor Documentation	2327
6.758.3.1 SugarBlock_3_VVV()	2327
6.758.4 Member Function Documentation	2327
6.758.4.1 operator[]()	2328
6.758.4.2 size()	2328
6.758.5 Member Data Documentation	2328
6.758.5.1 ptr	2328
6.758.5.2 x	2329
6.758.5.3 y	2329
6.758.5.4 z	2329
6.759 Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr > Class Template Reference	2330
6.759.1 Detailed Description	2330
6.759.2 Member Typedef Documentation	2331
6.759.2.1 VEC_TYPE	2331
6.759.3 Constructor & Destructor Documentation	2331
6.759.3.1 SugarComplex()	2331
6.759.4 Member Function Documentation	2331
6.759.4.1 operator[]()	2331
6.759.4.2 size()	2332
6.759.5 Member Data Documentation	2332
6.759.5.1 ptr	2332
6.759.5.2 vec	2332
6.760 Rcpp::sugar::SugarIterator< T > Class Template Reference	2333
6.760.1 Detailed Description	2333
6.760.2 Member Typedef Documentation	2334
6.760.2.1 difference_type	2334
6.760.2.2 iterator	2334
6.760.2.3 iterator_category	2334
6.760.2.4 pointer	2334

---

6.760.2.5 reference	2334
6.760.2.6 STORAGE_TYPE	2335
6.760.3 Constructor & Destructor Documentation	2335
6.760.3.1 SugarIterator() [1/3]	2335
6.760.3.2 SugarIterator() [2/3]	2335
6.760.3.3 SugarIterator() [3/3]	2335
6.760.4 Member Function Documentation	2335
6.760.4.1 operator!=(())	2336
6.760.4.2 operator*()	2336
6.760.4.3 operator+()	2336
6.760.4.4 operator++() [1/2]	2336
6.760.4.5 operator++() [2/2]	2337
6.760.4.6 operator+=(())	2337
6.760.4.7 operator-() [1/2]	2337
6.760.4.8 operator-() [2/2]	2337
6.760.4.9 operator--() [1/2]	2338
6.760.4.10 operator--() [2/2]	2338
6.760.4.11 operator==(())	2338
6.760.4.12 operator->()	2338
6.760.4.13 operator<()	2339
6.760.4.14 operator<=()	2339
6.760.4.15 operator==(())	2339
6.760.4.16 operator>()	2339
6.760.4.17 operator>=()	2340
6.760.4.18 operator[]()	2340
6.760.5 Member Data Documentation	2340
6.760.5.1 index	2340
6.760.5.2 ref	2341
6.761 Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr > Class Template Reference	2341
6.761.1 Detailed Description	2342
6.761.2 Member Typedef Documentation	2342
6.761.2.1 VEC_TYPE	2342
6.761.3 Constructor & Destructor Documentation	2342
6.761.3.1 SugarMath_1()	2342
6.761.4 Member Function Documentation	2343
6.761.4.1 operator[]()	2343
6.761.4.2 size()	2343
6.761.5 Member Data Documentation	2343
6.761.5.1 ptr	2344

---

6.761.5.2 vec	2344
6.762 Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr > Class Template Reference	2344
6.762.1 Detailed Description	2345
6.762.2 Member Typedef Documentation	2345
6.762.2.1 VEC_TYPE	2345
6.762.3 Constructor & Destructor Documentation	2346
6.762.3.1 SugarMath_1()	2346
6.762.4 Member Function Documentation	2346
6.762.4.1 operator[]()	2346
6.762.4.2 size()	2346
6.762.5 Member Data Documentation	2347
6.762.5.1 ptr	2347
6.762.5.2 vec	2347
6.763 Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr > Class Template Reference	2347
6.763.1 Detailed Description	2348
6.763.2 Member Typedef Documentation	2348
6.763.2.1 VEC_TYPE	2348
6.763.3 Constructor & Destructor Documentation	2348
6.763.3.1 SugarMath_1()	2349
6.763.4 Member Function Documentation	2349
6.763.4.1 operator[]()	2349
6.763.4.2 size()	2350
6.763.5 Member Data Documentation	2350
6.763.5.1 ptr	2350
6.763.5.2 vec	2350
6.764 Rcpp::sugar::Sum< RTYPE, NA, T > Class Template Reference	2351
6.764.1 Detailed Description	2352
6.764.2 Member Typedef Documentation	2352
6.764.2.1 STORAGE	2352
6.764.2.2 VEC_EXT	2352
6.764.2.3 VEC_TYPE	2352
6.764.3 Constructor & Destructor Documentation	2352
6.764.3.1 Sum()	2353
6.764.4 Member Function Documentation	2353
6.764.4.1 get()	2353
6.764.5 Member Data Documentation	2353
6.764.5.1 object	2353
6.765 Rcpp::sugar::Sum< REALSXP, NA, T > Class Template Reference	2354
6.765.1 Detailed Description	2355

---

6.765.2 Member Typedef Documentation	2355
6.765.2.1 VEC_EXT	2355
6.765.2.2 VEC_TYPE	2355
6.765.3 Constructor & Destructor Documentation	2355
6.765.3.1 Sum()	2355
6.765.4 Member Function Documentation	2356
6.765.4.1 get()	2356
6.765.5 Member Data Documentation	2356
6.765.5.1 object	2356
6.766 Rcpp::sugar::Sum< RTYPE, false, T > Class Template Reference	2356
6.766.1 Detailed Description	2357
6.766.2 Member Typedef Documentation	2357
6.766.2.1 STORAGE	2357
6.766.2.2 VEC_EXT	2358
6.766.2.3 VEC_TYPE	2358
6.766.3 Constructor & Destructor Documentation	2358
6.766.3.1 Sum()	2358
6.766.4 Member Function Documentation	2358
6.766.4.1 get()	2358
6.766.5 Member Data Documentation	2358
6.766.5.1 object	2359
6.767 Rcpp::SuspendRNGSynchronizationScope Class Reference	2359
6.767.1 Detailed Description	2359
6.767.2 Constructor & Destructor Documentation	2359
6.767.2.1 SuspendRNGSynchronizationScope()	2359
6.767.2.2 ~SuspendRNGSynchronizationScope()	2360
6.768 Rcpp::sugar::Table< RTYPE, TABLE_T > Class Template Reference	2360
6.768.1 Detailed Description	2361
6.768.2 Member Typedef Documentation	2361
6.768.2.1 HASH	2361
6.768.2.2 Inserter	2361
6.768.2.3 SORTED_MAP	2361
6.768.2.4 STORAGE	2362
6.768.3 Constructor & Destructor Documentation	2362
6.768.3.1 Table()	2362
6.768.4 Member Function Documentation	2362
6.768.4.1 operator IntegerVector()	2363
6.768.5 Member Data Documentation	2363
6.768.5.1 hash	2363

---

6.768.5.2 map	2363
6.769 Rcpp::TagProxyPolicy< XPtrClass >::TagProxy Class Reference	2364
6.769.1 Detailed Description	2365
6.769.2 Constructor & Destructor Documentation	2365
6.769.2.1 TagProxy()	2365
6.769.3 Member Function Documentation	2365
6.769.3.1 get()	2365
6.769.3.2 operator SEXP()	2366
6.769.3.3 operator U()	2366
6.769.3.4 operator=() [1/2]	2366
6.769.3.5 operator=() [2/2]	2367
6.769.3.6 set()	2367
6.769.4 Member Data Documentation	2367
6.769.4.1 xp	2367
6.770 Rcpp::TagProxyPolicy< XPtrClass > Class Template Reference	2367
6.770.1 Detailed Description	2368
6.770.2 Member Function Documentation	2368
6.770.2.1 tag() [1/2]	2368
6.770.2.2 tag() [2/2]	2368
6.771 Rcpp::sugar::Tail< RTYPE, NA, T > Class Template Reference	2369
6.771.1 Detailed Description	2370
6.771.2 Member Typedef Documentation	2370
6.771.2.1 STORAGE	2370
6.771.2.2 VEC_TYPE	2370
6.771.3 Constructor & Destructor Documentation	2370
6.771.3.1 Tail()	2370
6.771.4 Member Function Documentation	2371
6.771.4.1 operator[]()	2371
6.771.4.2 size()	2371
6.771.5 Member Data Documentation	2371
6.771.5.1 n	2371
6.771.5.2 object	2371
6.771.5.3 start	2372
6.772 Rcpp::stats::TGenerator Class Reference	2372
6.772.1 Detailed Description	2373
6.772.2 Constructor & Destructor Documentation	2373
6.772.2.1 TGenerator()	2373
6.772.3 Member Function Documentation	2373
6.772.3.1 operator()()	2373

---

6.772.4 Member Data Documentation	2374
6.772.4.1 df	2374
6.772.4.2 df_2	2374
6.773 Rcpp::Timer Class Reference	2374
6.773.1 Detailed Description	2375
6.773.2 Member Typedef Documentation	2375
6.773.2.1 Step	2375
6.773.2.2 Steps	2375
6.773.3 Constructor & Destructor Documentation	2375
6.773.3.1 Timer() [1/2]	2375
6.773.3.2 Timer() [2/2]	2376
6.773.4 Member Function Documentation	2376
6.773.4.1 get_timers()	2376
6.773.4.2 now()	2376
6.773.4.3 operator SEXP()	2377
6.773.4.4 origin()	2377
6.773.4.5 step()	2377
6.773.5 Member Data Documentation	2378
6.773.5.1 data	2378
6.773.5.2 start_time	2378
6.774 Timer Class Reference	2378
6.774.1 Detailed Description	2379
6.774.2 Constructor & Destructor Documentation	2379
6.774.2.1 Timer()	2379
6.774.3 Member Function Documentation	2379
6.774.3.1 CumulativeTime()	2380
6.774.3.2 ElapsedTime()	2380
6.774.3.3 getFractionalSeconds()	2380
6.774.3.4 Reset()	2381
6.774.3.5 Start()	2381
6.774.3.6 Stop()	2382
6.774.4 Member Data Documentation	2382
6.774.4.1 cumul	2382
6.774.4.2 elapsed	2383
6.774.4.3 end_t	2383
6.774.4.4 start_t	2383
6.774.4.5 sys_time	2384
6.775 Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T > Class Template Reference	2384
6.775.1 Detailed Description	2385

---

6.775.2 Member Typedef Documentation	2386
6.775.2.1 EXT	2386
6.775.2.2 STORAGE	2386
6.775.2.3 VEC_TYPE	2386
6.775.3 Constructor & Destructor Documentation	2386
6.775.3.1 Times_Vector_Primitive()	2386
6.775.4 Member Function Documentation	2387
6.775.4.1 operator[]()	2387
6.775.4.2 size()	2387
6.775.5 Member Data Documentation	2387
6.775.5.1 lhs	2387
6.775.5.2 rhs	2388
6.775.5.3 rhs_na	2388
6.776 Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T > Class Template Reference	2388
6.776.1 Detailed Description	2390
6.776.2 Member Typedef Documentation	2390
6.776.2.1 EXT	2390
6.776.2.2 VEC_TYPE	2390
6.776.3 Constructor & Destructor Documentation	2391
6.776.3.1 Times_Vector_Primitive()	2391
6.776.4 Member Function Documentation	2391
6.776.4.1 operator[]()	2391
6.776.4.2 size()	2391
6.776.5 Member Data Documentation	2391
6.776.5.1 lhs	2392
6.776.5.2 rhs	2392
6.777 Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T > Class Template Reference	2392
6.777.1 Detailed Description	2393
6.777.2 Member Typedef Documentation	2394
6.777.2.1 EXT	2394
6.777.2.2 VEC_TYPE	2394
6.777.3 Constructor & Destructor Documentation	2394
6.777.3.1 Times_Vector_Primitive()	2394
6.777.4 Member Function Documentation	2394
6.777.4.1 operator[]()	2395
6.777.4.2 size()	2395
6.777.5 Member Data Documentation	2395
6.777.5.1 lhs	2395
6.777.5.2 rhs	2395



---

6.778 Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T > Class Template Reference . . . . .	2396
6.778.1 Detailed Description . . . . .	2397
6.778.2 Member Typedef Documentation . . . . .	2398
6.778.2.1 EXT . . . . .	2398
6.778.2.2 STORAGE . . . . .	2398
6.778.2.3 VEC_TYPE . . . . .	2398
6.778.3 Constructor & Destructor Documentation . . . . .	2398
6.778.3.1 Times_Vector_Primitive() . . . . .	2398
6.778.4 Member Function Documentation . . . . .	2399
6.778.4.1 operator[]() . . . . .	2399
6.778.4.2 size() . . . . .	2399
6.778.5 Member Data Documentation . . . . .	2399
6.778.5.1 lhs . . . . .	2399
6.778.5.2 rhs . . . . .	2399
6.778.5.3 rhs_na . . . . .	2400
6.779 Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T > Class Template Reference . . . . .	2400
6.779.1 Detailed Description . . . . .	2401
6.779.2 Member Typedef Documentation . . . . .	2402
6.779.2.1 EXT . . . . .	2402
6.779.2.2 STORAGE . . . . .	2402
6.779.2.3 VEC_TYPE . . . . .	2402
6.779.3 Constructor & Destructor Documentation . . . . .	2402
6.779.3.1 Times_Vector_Primitive_nona() . . . . .	2402
6.779.4 Member Function Documentation . . . . .	2403
6.779.4.1 operator[]() . . . . .	2403
6.779.4.2 size() . . . . .	2403
6.779.5 Member Data Documentation . . . . .	2403
6.779.5.1 lhs . . . . .	2403
6.779.5.2 rhs . . . . .	2404
6.780 Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T > Class Template Reference . . . . .	2404
6.780.1 Detailed Description . . . . .	2405
6.780.2 Member Typedef Documentation . . . . .	2405
6.780.2.1 EXT . . . . .	2405
6.780.2.2 VEC_TYPE . . . . .	2405
6.780.3 Constructor & Destructor Documentation . . . . .	2405
6.780.3.1 Times_Vector_Primitive_nona() . . . . .	2406
6.780.4 Member Function Documentation . . . . .	2406
6.780.4.1 operator[]() . . . . .	2406
6.780.4.2 size() . . . . .	2406

---

6.780.5 Member Data Documentation	2406
6.780.5.1 lhs	2406
6.780.5.2 rhs	2407
6.781 Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T > Class Template Reference	2407
6.781.1 Detailed Description	2408
6.781.2 Member Typedef Documentation	2409
6.781.2.1 EXT	2409
6.781.2.2 VEC_TYPE	2409
6.781.3 Constructor & Destructor Documentation	2409
6.781.3.1 Times_Vector_Primitive_nona()	2409
6.781.4 Member Function Documentation	2409
6.781.4.1 operator[]()	2410
6.781.4.2 size()	2410
6.781.5 Member Data Documentation	2410
6.781.5.1 lhs	2410
6.781.5.2 rhs	2410
6.782 Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T > Class Template Reference	2411
6.782.1 Detailed Description	2412
6.782.2 Member Typedef Documentation	2413
6.782.2.1 EXT	2413
6.782.2.2 STORAGE	2413
6.782.2.3 VEC_TYPE	2413
6.782.3 Constructor & Destructor Documentation	2413
6.782.3.1 Times_Vector_Primitive_nona()	2413
6.782.4 Member Function Documentation	2414
6.782.4.1 operator[]()	2414
6.782.4.2 size()	2414
6.782.5 Member Data Documentation	2414
6.782.5.1 lhs	2414
6.782.5.2 rhs	2414
6.783 Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	2415
6.783.1 Detailed Description	2416
6.783.2 Member Typedef Documentation	2416
6.783.2.1 LHS_EXT	2416
6.783.2.2 LHS_TYPE	2416
6.783.2.3 RHS_EXT	2416
6.783.2.4 RHS_TYPE	2417
6.783.2.5 STORAGE	2417

6.783.3 Constructor & Destructor Documentation	2417
6.783.3.1 Times_Vector_Vector()	2417
6.783.4 Member Function Documentation	2417
6.783.4.1 operator[]()	2417
6.783.4.2 size()	2418
6.783.5 Member Data Documentation	2418
6.783.5.1 lhs	2418
6.783.5.2 rhs	2418
6.784 Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > Class Template Reference	2419
6.784.1 Detailed Description	2420
6.784.2 Member Typedef Documentation	2420
6.784.2.1 LHS_EXT	2420
6.784.2.2 LHS_TYPE	2420
6.784.2.3 RHS_EXT	2420
6.784.2.4 RHS_TYPE	2421
6.784.3 Constructor & Destructor Documentation	2421
6.784.3.1 Times_Vector_Vector()	2421
6.784.4 Member Function Documentation	2421
6.784.4.1 operator[]()	2421
6.784.4.2 size()	2421
6.784.5 Member Data Documentation	2422
6.784.5.1 lhs	2422
6.784.5.2 rhs	2422
6.785 Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > Class Template Reference	2422
6.785.1 Detailed Description	2423
6.785.2 Member Typedef Documentation	2423
6.785.2.1 LHS_EXT	2423
6.785.2.2 LHS_TYPE	2424
6.785.2.3 RHS_EXT	2424
6.785.2.4 RHS_TYPE	2424
6.785.3 Constructor & Destructor Documentation	2424
6.785.3.1 Times_Vector_Vector()	2424
6.785.4 Member Function Documentation	2424
6.785.4.1 operator[]()	2425
6.785.4.2 size()	2425
6.785.5 Member Data Documentation	2425
6.785.5.1 lhs	2425
6.785.5.2 rhs	2425

---

6.786 Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > Class Template Reference	2426
6.786.1 Detailed Description	2427
6.786.2 Member Typedef Documentation	2427
6.786.2.1 LHS_EXT	2427
6.786.2.2 LHS_TYPE	2427
6.786.2.3 RHS_EXT	2427
6.786.2.4 RHS_TYPE	2428
6.786.3 Constructor & Destructor Documentation	2428
6.786.3.1 Times_Vector_Vector()	2428
6.786.4 Member Function Documentation	2428
6.786.4.1 operator[]()	2428
6.786.4.2 size()	2428
6.786.5 Member Data Documentation	2429
6.786.5.1 lhs	2429
6.786.5.2 rhs	2429
6.787 Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	2429
6.787.1 Detailed Description	2430
6.787.2 Member Typedef Documentation	2430
6.787.2.1 LHS_EXT	2430
6.787.2.2 LHS_TYPE	2431
6.787.2.3 RHS_EXT	2431
6.787.2.4 RHS_TYPE	2431
6.787.3 Constructor & Destructor Documentation	2431
6.787.3.1 Times_Vector_Vector()	2431
6.787.4 Member Function Documentation	2431
6.787.4.1 operator[]()	2432
6.787.4.2 size()	2432
6.787.5 Member Data Documentation	2432
6.787.5.1 lhs	2432
6.787.5.2 rhs	2432
6.788 Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > Class Template Reference	2433
6.788.1 Detailed Description	2434
6.788.2 Member Typedef Documentation	2434
6.788.2.1 LHS_EXT	2434
6.788.2.2 LHS_TYPE	2434
6.788.2.3 RHS_EXT	2434
6.788.2.4 RHS_TYPE	2435
6.788.2.5 STORAGE	2435

---

6.788.3 Constructor & Destructor Documentation	2435
6.788.3.1 Times_Vector_Vector()	2435
6.788.4 Member Function Documentation	2435
6.788.4.1 operator[]()	2435
6.788.4.2 size()	2436
6.788.5 Member Data Documentation	2436
6.788.5.1 lhs	2436
6.788.5.2 rhs	2436
6.789 Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > Class Template Reference	2437
6.789.1 Detailed Description	2438
6.789.2 Member Typedef Documentation	2438
6.789.2.1 LHS_EXT	2438
6.789.2.2 LHS_TYPE	2438
6.789.2.3 RHS_EXT	2438
6.789.2.4 RHS_TYPE	2439
6.789.2.5 STORAGE	2439
6.789.3 Constructor & Destructor Documentation	2439
6.789.3.1 Times_Vector_Vector()	2439
6.789.4 Member Function Documentation	2439
6.789.4.1 operator[]()	2439
6.789.4.2 size()	2440
6.789.5 Member Data Documentation	2440
6.789.5.1 lhs	2440
6.789.5.2 rhs	2440
6.790 Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > Class Template Reference	2441
6.790.1 Detailed Description	2442
6.790.2 Member Typedef Documentation	2442
6.790.2.1 LHS_EXT	2442
6.790.2.2 LHS_TYPE	2442
6.790.2.3 RHS_EXT	2442
6.790.2.4 RHS_TYPE	2443
6.790.2.5 STORAGE	2443
6.790.3 Constructor & Destructor Documentation	2443
6.790.3.1 Times_Vector_Vector()	2443
6.790.4 Member Function Documentation	2443
6.790.4.1 operator[]()	2443
6.790.4.2 size()	2444
6.790.5 Member Data Documentation	2444

---

6.790.5.1 lhs	2444
6.790.5.2 rhs	2444
6.791 Rcpp::ttinfo Struct Reference	2444
6.791.1 Detailed Description	2445
6.791.2 Member Data Documentation	2445
6.791.2.1 tt_abbrind	2445
6.791.2.2 tt_gmtoff	2445
6.791.2.3 tt_isdst	2445
6.791.2.4 tt_tisgmt	2445
6.791.2.5 tt_tisstd	2446
6.792 Rcpp::attributes::Type Class Reference	2446
6.792.1 Detailed Description	2446
6.792.2 Constructor & Destructor Documentation	2446
6.792.2.1 Type() [1/2]	2447
6.792.2.2 Type() [2/2]	2447
6.792.3 Member Function Documentation	2447
6.792.3.1 empty()	2447
6.792.3.2 full_name()	2448
6.792.3.3 isConst()	2448
6.792.3.4 isReference()	2449
6.792.3.5 isVoid()	2449
6.792.3.6 name()	2449
6.792.3.7 operator"!="()	2450
6.792.3.8 operator==( )	2450
6.792.4 Member Data Documentation	2450
6.792.4.1 isConst_	2450
6.792.4.2 isReference_	2450
6.792.4.3 name_	2451
6.793 Rcpp::tzhead Struct Reference	2451
6.793.1 Detailed Description	2451
6.793.2 Member Data Documentation	2451
6.793.2.1 tzh_charcnt	2451
6.793.2.2 tzh_leapcnt	2452
6.793.2.3 tzh_magic	2452
6.793.2.4 tzh_reserved	2452
6.793.2.5 tzh_timecnt	2452
6.793.2.6 tzh_tisgmtcnt	2452
6.793.2.7 tzh_tisstdcnt	2452
6.793.2.8 tzh_typecnt	2453

---

6.793.2.9 <code>tzh_version</code> . . . . .	2453
6.794 <code>Rcpp::traits::un_pointer&lt; T &gt;</code> Struct Template Reference . . . . .	2453
6.794.1 Detailed Description . . . . .	2453
6.794.2 Member Typedef Documentation . . . . .	2453
6.794.2.1 <code>type</code> . . . . .	2454
6.795 <code>Rcpp::traits::un_pointer&lt; object&lt; T &gt; &gt;</code> Struct Template Reference . . . . .	2454
6.795.1 Detailed Description . . . . .	2454
6.795.2 Member Typedef Documentation . . . . .	2454
6.795.2.1 <code>type</code> . . . . .	2454
6.796 <code>Rcpp::traits::un_pointer&lt; T * &gt;</code> Struct Template Reference . . . . .	2455
6.796.1 Detailed Description . . . . .	2455
6.796.2 Member Typedef Documentation . . . . .	2455
6.796.2.1 <code>type</code> . . . . .	2455
6.797 <code>Rcpp::unary_call&lt; T, RESULT_TYPE &gt;</code> Class Template Reference . . . . .	2456
6.797.1 Detailed Description . . . . .	2457
6.797.2 Constructor & Destructor Documentation . . . . .	2457
6.797.2.1 <code>unary_call()</code> [1/3] . . . . .	2457
6.797.2.2 <code>unary_call()</code> [2/3] . . . . .	2457
6.797.2.3 <code>unary_call()</code> [3/3] . . . . .	2457
6.797.3 Member Function Documentation . . . . .	2458
6.797.3.1 <code>operator&gt;()</code> . . . . .	2458
6.797.4 Member Data Documentation . . . . .	2458
6.797.4.1 <code>call</code> . . . . .	2458
6.797.4.2 <code>proxy</code> . . . . .	2458
6.798 <code>Rcpp::sugar::unary_minus&lt; RTYPE, NA &gt;</code> Class Template Reference . . . . .	2459
6.798.1 Detailed Description . . . . .	2459
6.798.2 Member Typedef Documentation . . . . .	2459
6.798.2.1 <code>RESULT</code> . . . . .	2459
6.798.2.2 <code>STORAGE</code> . . . . .	2459
6.798.3 Member Function Documentation . . . . .	2460
6.798.3.1 <code>apply()</code> . . . . .	2460
6.799 <code>Rcpp::sugar::unary_minus&lt; CPLXSCP, false &gt;</code> Class Reference . . . . .	2460
6.799.1 Detailed Description . . . . .	2460
6.799.2 Member Function Documentation . . . . .	2460
6.799.2.1 <code>apply()</code> . . . . .	2460
6.800 <code>Rcpp::sugar::unary_minus&lt; CPLXSCP, NA &gt;</code> Class Template Reference . . . . .	2461
6.800.1 Detailed Description . . . . .	2461
6.800.2 Member Function Documentation . . . . .	2461
6.800.2.1 <code>apply()</code> . . . . .	2461

---

6.801 Rcpp::sugar::unary_minus< RTYPE, false > Class Template Reference . . . . .	2462
6.801.1 Detailed Description . . . . .	2462
6.801.2 Member Typedef Documentation . . . . .	2462
6.801.2.1 RESULT . . . . .	2462
6.801.2.2 STORAGE . . . . .	2462
6.801.3 Member Function Documentation . . . . .	2463
6.801.3.1 apply() . . . . .	2463
6.802 Rcpp::sugar::unary_minus_result_type< RTYPE > Struct Template Reference . . . . .	2463
6.802.1 Detailed Description . . . . .	2463
6.802.2 Member Typedef Documentation . . . . .	2463
6.802.2.1 type . . . . .	2463
6.802.3 Member Enumeration Documentation . . . . .	2464
6.802.3.1 anonymous enum . . . . .	2464
6.803 Rcpp::sugar::unary_minus_result_type< LGLSXP > Struct Reference . . . . .	2464
6.803.1 Detailed Description . . . . .	2464
6.803.2 Member Typedef Documentation . . . . .	2464
6.803.2.1 type . . . . .	2464
6.803.3 Member Enumeration Documentation . . . . .	2465
6.803.3.1 anonymous enum . . . . .	2465
6.804 Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T > Class Template Reference . . . . .	2465
6.804.1 Detailed Description . . . . .	2466
6.804.2 Member Typedef Documentation . . . . .	2466
6.804.2.1 OPERATOR . . . . .	2466
6.804.2.2 RESULT . . . . .	2466
6.804.2.3 STORAGE . . . . .	2467
6.804.2.4 VEC_TYPE . . . . .	2467
6.804.3 Constructor & Destructor Documentation . . . . .	2467
6.804.3.1 UnaryMinus_Vector() . . . . .	2467
6.804.4 Member Function Documentation . . . . .	2467
6.804.4.1 operator[]() . . . . .	2468
6.804.4.2 size() . . . . .	2468
6.804.5 Member Data Documentation . . . . .	2469
6.804.5.1 lhs . . . . .	2469
6.804.5.2 op . . . . .	2469
6.805 Rcpp::stats::UnifGenerator Class Reference . . . . .	2469
6.805.1 Detailed Description . . . . .	2470
6.805.2 Constructor & Destructor Documentation . . . . .	2470
6.805.2.1 UnifGenerator() . . . . .	2470
6.805.3 Member Function Documentation . . . . .	2471



---

6.805.3.1 operator>()	2471
6.805.4 Member Data Documentation	2471
6.805.4.1 diff	2471
6.805.4.2 min	2471
6.806 Rcpp::stats::UnifGenerator__0__1 Class Reference	2472
6.806.1 Detailed Description	2472
6.806.2 Constructor & Destructor Documentation	2473
6.806.2.1 UnifGenerator__0__1()	2473
6.806.3 Member Function Documentation	2473
6.806.3.1 operator>()	2473
6.807 Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Class Template Reference	2473
6.807.1 Detailed Description	2474
6.807.2 Member Typedef Documentation	2474
6.807.2.1 ITERATOR	2474
6.807.2.2 SET	2475
6.807.2.3 STORAGE	2475
6.807.3 Constructor & Destructor Documentation	2475
6.807.3.1 Union()	2475
6.807.4 Member Function Documentation	2476
6.807.4.1 get()	2476
6.807.5 Member Data Documentation	2476
6.807.5.1 result	2476
6.808 Rcpp::internal::UnwindData Struct Reference	2477
6.808.1 Detailed Description	2477
6.808.2 Member Data Documentation	2477
6.808.2.1 jmpbuf	2477
6.809 Rcpp::sugar::UpperTri< RTYPE, NA, T > Class Template Reference	2477
6.809.1 Detailed Description	2479
6.809.2 Member Typedef Documentation	2479
6.809.2.1 MatBase	2479
6.809.2.2 Method	2480
6.809.3 Constructor & Destructor Documentation	2480
6.809.3.1 UpperTri()	2480
6.809.4 Member Function Documentation	2480
6.809.4.1 get()	2480
6.809.4.2 get_diag_false()	2480
6.809.4.3 get_diag_true()	2481
6.809.4.4 ncol()	2481
6.809.4.5 nrow()	2481

---

6.809.4.6 operator>()	2481
6.809.4.7 size()	2482
6.809.5 Member Data Documentation	2482
6.809.5.1 getter	2482
6.809.5.2 nc	2482
6.809.5.3 nr	2482
6.810 Rcpp::sugar::Var< RTYPE, NA, T > Class Template Reference	2483
6.810.1 Detailed Description	2484
6.810.2 Member Typedef Documentation	2484
6.810.2.1 VEC_TYPE	2484
6.810.3 Constructor & Destructor Documentation	2484
6.810.3.1 Var()	2484
6.810.4 Member Function Documentation	2484
6.810.4.1 get()	2485
6.810.5 Member Data Documentation	2485
6.810.5.1 object	2485
6.811 Rcpp::sugar::Var< CPLXSP, NA, T > Class Template Reference	2486
6.811.1 Detailed Description	2487
6.811.2 Member Typedef Documentation	2487
6.811.2.1 VEC_TYPE	2487
6.811.3 Constructor & Destructor Documentation	2487
6.811.3.1 Var()	2487
6.811.4 Member Function Documentation	2487
6.811.4.1 get()	2488
6.811.5 Member Data Documentation	2488
6.811.5.1 object	2488
6.812 Vec Class Reference	2489
6.812.1 Detailed Description	2489
6.812.2 Member Typedef Documentation	2490
6.812.2.1 proxy [1/2]	2490
6.812.2.2 proxy [2/2]	2490
6.812.3 Constructor & Destructor Documentation	2490
6.812.3.1 Vec() [1/3]	2490
6.812.3.2 Vec() [2/3]	2490
6.812.3.3 Vec() [3/3]	2491
6.812.4 Member Function Documentation	2491
6.812.4.1 operator[]() [1/5]	2491
6.812.4.2 operator[]() [2/5]	2491
6.812.4.3 operator[]() [3/5]	2492

---

6.812.4.4 operator[]() [4/5]	2492
6.812.4.5 operator[]() [5/5]	2493
6.812.5 Member Data Documentation	2493
6.812.5.1 cache	2493
6.812.5.2 data	2493
6.813 Rcpp::Vector< RTYPE, StoragePolicy > Class Template Reference	2494
6.813.1 Detailed Description	2498
6.813.2 Member Typedef Documentation	2499
6.813.2.1 const_iterator	2499
6.813.2.2 const_Proxy	2499
6.813.2.3 converter_type	2499
6.813.2.4 Indexer	2499
6.813.2.5 init_type	2500
6.813.2.6 iterator	2500
6.813.2.7 NameProxy	2500
6.813.2.8 Proxy	2500
6.813.2.9 Storage	2500
6.813.2.10 stored_type	2501
6.813.2.11 value_type	2501
6.813.3 Constructor & Destructor Documentation	2501
6.813.3.1 Vector() [1/25]	2501
6.813.3.2 Vector() [2/25]	2502
6.813.3.3 Vector() [3/25]	2502
6.813.3.4 Vector() [4/25]	2502
6.813.3.5 Vector() [5/25]	2503
6.813.3.6 Vector() [6/25]	2503
6.813.3.7 Vector() [7/25]	2504
6.813.3.8 Vector() [8/25]	2504
6.813.3.9 Vector() [9/25]	2504
6.813.3.10 Vector() [10/25]	2505
6.813.3.11 Vector() [11/25]	2505
6.813.3.12 Vector() [12/25]	2506
6.813.3.13 Vector() [13/25]	2506
6.813.3.14 Vector() [14/25]	2507
6.813.3.15 Vector() [15/25]	2508
6.813.3.16 Vector() [16/25]	2508
6.813.3.17 Vector() [17/25]	2509
6.813.3.18 Vector() [18/25]	2509
6.813.3.19 Vector() [19/25]	2510

---

6.813.3.20	Vector()	[20/25]	2510
6.813.3.21	Vector()	[21/25]	2511
6.813.3.22	Vector()	[22/25]	2512
6.813.3.23	Vector()	[23/25]	2512
6.813.3.24	Vector()	[24/25]	2513
6.813.3.25	Vector()	[25/25]	2513
6.813.4	Member Function Documentation		2514
6.813.4.1	assign()		2514
6.813.4.2	assign_object()	[1/2]	2514
6.813.4.3	assign_object()	[2/2]	2515
6.813.4.4	assign_sugar_expression()		2516
6.813.4.5	at()	[1/2]	2516
6.813.4.6	at()	[2/2]	2517
6.813.4.7	begin()	[1/2]	2517
6.813.4.8	begin()	[2/2]	2518
6.813.4.9	cbegin()		2518
6.813.4.10	cend()		2518
6.813.4.11	containsElementNamed()		2519
6.813.4.12	create()		2519
6.813.4.13	dims()		2520
6.813.4.14	end()	[1/2]	2520
6.813.4.15	end()	[2/2]	2521
6.813.4.16	erase()	[1/4]	2521
6.813.4.17	erase()	[2/4]	2522
6.813.4.18	erase()	[3/4]	2522
6.813.4.19	erase()	[4/4]	2523
6.813.4.20	erase_range__impl()		2524
6.813.4.21	erase_single__impl()		2524
6.813.4.22	eval()	[1/2]	2525
6.813.4.23	eval()	[2/2]	2526
6.813.4.24	fill()		2526
6.813.4.25	fill__dispatch()	[1/2]	2527
6.813.4.26	fill__dispatch()	[2/2]	2527
6.813.4.27	fill_or_generate()		2528
6.813.4.28	fill_or_generate__impl()	[1/2]	2529
6.813.4.29	fill_or_generate__impl()	[2/2]	2529
6.813.4.30	findName()		2530
6.813.4.31	get_na()		2530
6.813.4.32	import()		2530

---

6.813.4.33 import_expression()	2531
6.813.4.34 import_sugar_expression() [1/2]	2531
6.813.4.35 import_sugar_expression() [2/2]	2532
6.813.4.36 import_transform()	2533
6.813.4.37 init()	2533
6.813.4.38 insert() [1/2]	2534
6.813.4.39 insert() [2/2]	2534
6.813.4.40 insert__impl() [1/2]	2535
6.813.4.41 insert__impl() [2/2]	2536
6.813.4.42 is_na()	2537
6.813.4.43 length()	2537
6.813.4.44 offset() [1/3]	2538
6.813.4.45 offset() [2/3]	2538
6.813.4.46 offset() [3/3]	2539
6.813.4.47 operator RObject()	2539
6.813.4.48 operator>() [1/6]	2539
6.813.4.49 operator>() [2/6]	2540
6.813.4.50 operator>() [3/6]	2540
6.813.4.51 operator>() [4/6]	2541
6.813.4.52 operator>() [5/6]	2541
6.813.4.53 operator>() [6/6]	2541
6.813.4.54 operator+>() [1/2]	2542
6.813.4.55 operator+>() [2/2]	2542
6.813.4.56 operator=()	2543
6.813.4.57 operator=() [2/2]	2543
6.813.4.58 operator[]()	2544
6.813.4.59 operator[]() [2/7]	2544
6.813.4.60 operator[]() [3/7]	2544
6.813.4.61 operator[]() [4/7]	2544
6.813.4.62 operator[]() [5/7]	2545
6.813.4.63 operator[]() [6/7]	2545
6.813.4.64 operator[]() [7/7]	2545
6.813.4.65 push_back() [1/2]	2546
6.813.4.66 push_back() [2/2]	2546
6.813.4.67 push_back__impl() [1/2]	2547
6.813.4.68 push_back__impl() [2/2]	2548
6.813.4.69 push_back_name__impl() [1/2]	2549
6.813.4.70 push_back_name__impl() [2/2]	2550
6.813.4.71 push_front() [1/2]	2551

---

6.813.4.72 push_front() [2/2]	2552
6.813.4.73 push_front__impl() [1/2]	2552
6.813.4.74 push_front__impl() [2/2]	2553
6.813.4.75 push_front_name__impl() [1/2]	2554
6.813.4.76 push_front_name__impl() [2/2]	2555
6.813.4.77 replace_element()	2556
6.813.4.78 replace_element__dispatch() [1/2]	2557
6.813.4.79 replace_element__dispatch() [2/2]	2557
6.813.4.80 replace_element__dispatch__isArgument() [1/2]	2558
6.813.4.81 replace_element__dispatch__isArgument() [2/2]	2559
6.813.4.82 size()	2559
6.813.4.83 sort()	2560
6.813.4.84 update()	2560
6.813.5 Member Data Documentation	2561
6.813.5.1 cache	2561
6.814 Rcpp::VectorBase< RTYPE, na, VECTOR > Class Template Reference	2561
6.814.1 Detailed Description	2563
6.814.2 Member Typedef Documentation	2563
6.814.2.1 const_iterator	2563
6.814.2.2 elem_type	2564
6.814.2.3 iterator	2564
6.814.2.4 stored_type	2564
6.814.3 Member Function Documentation	2564
6.814.3.1 begin()	2564
6.814.3.2 cbegin()	2564
6.814.3.3 cend()	2565
6.814.3.4 end()	2565
6.814.3.5 get_ref() [1/2]	2566
6.814.3.6 get_ref() [2/2]	2566
6.814.3.7 operator[]()	2566
6.814.3.8 size()	2566
6.815 Rcpp::sugar::Vectorized< Func, NA, VEC > Class Template Reference	2567
6.815.1 Detailed Description	2569
6.815.2 Member Typedef Documentation	2569
6.815.2.1 VEC_EXT	2569
6.815.2.2 VEC_TYPE	2569
6.815.3 Constructor & Destructor Documentation	2569
6.815.3.1 Vectorized()	2570
6.815.4 Member Function Documentation	2570

---

6.815.4.1 operator[]()	2570
6.815.4.2 size()	2570
6.815.5 Member Data Documentation	2570
6.815.5.1 object	2570
6.816 Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC > Class Template Reference	2571
6.816.1 Detailed Description	2572
6.816.2 Member Typedef Documentation	2573
6.816.2.1 VEC_EXT	2573
6.816.2.2 VEC_TYPE	2573
6.816.3 Constructor & Destructor Documentation	2573
6.816.3.1 Vectorized_INTSXP()	2573
6.816.4 Member Function Documentation	2573
6.816.4.1 operator[]()	2573
6.816.4.2 size()	2574
6.816.5 Member Data Documentation	2574
6.816.5.1 object	2574
6.817 Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC > Class Template Reference	2574
6.817.1 Detailed Description	2575
6.817.2 Member Typedef Documentation	2576
6.817.2.1 VEC_EXT	2576
6.817.2.2 VEC_TYPE	2576
6.817.3 Constructor & Destructor Documentation	2576
6.817.3.1 Vectorized_INTSXP()	2576
6.817.4 Member Function Documentation	2576
6.817.4.1 operator[]()	2576
6.817.4.2 size()	2577
6.817.5 Member Data Documentation	2577
6.817.5.1 object	2577
6.818 Rcpp::void_type Struct Reference	2577
6.818.1 Detailed Description	2577
6.819 Rcpp::traits::void_wrap_tag Struct Reference	2577
6.819.1 Detailed Description	2578
6.820 Rcpp::stats::WeibullGenerator Class Reference	2578
6.820.1 Detailed Description	2579
6.820.2 Constructor & Destructor Documentation	2579
6.820.2.1 WeibullGenerator()	2579
6.820.3 Member Function Documentation	2579
6.820.3.1 operator()()	2579
6.820.4 Member Data Documentation	2580

---

6.820.4.1 scale	2580
6.820.4.2 shape_inv	2580
6.821 Rcpp::stats::WeibullGenerator__scale1 Class Reference	2580
6.821.1 Detailed Description	2581
6.821.2 Constructor & Destructor Documentation	2581
6.821.2.1 WeibullGenerator__scale1()	2581
6.821.3 Member Function Documentation	2582
6.821.3.1 operator>()	2582
6.821.4 Member Data Documentation	2582
6.821.4.1 shape_inv	2582
6.822 Rcpp::sugar::WhichMax< RTYPE, NA, T > Class Template Reference	2583
6.822.1 Detailed Description	2583
6.822.2 Member Typedef Documentation	2584
6.822.2.1 STORAGE	2584
6.822.2.2 VEC_TYPE	2584
6.822.3 Constructor & Destructor Documentation	2584
6.822.3.1 WhichMax()	2584
6.822.4 Member Function Documentation	2584
6.822.4.1 get()	2585
6.822.5 Member Data Documentation	2585
6.822.5.1 obj	2585
6.823 Rcpp::sugar::WhichMax< RTYPE, false, T > Class Template Reference	2586
6.823.1 Detailed Description	2587
6.823.2 Member Typedef Documentation	2587
6.823.2.1 STORAGE	2587
6.823.2.2 VEC_TYPE	2587
6.823.3 Constructor & Destructor Documentation	2587
6.823.3.1 WhichMax()	2587
6.823.4 Member Function Documentation	2587
6.823.4.1 get()	2588
6.823.5 Member Data Documentation	2588
6.823.5.1 obj	2588
6.824 Rcpp::sugar::WhichMin< RTYPE, NA, T > Class Template Reference	2589
6.824.1 Detailed Description	2589
6.824.2 Member Typedef Documentation	2590
6.824.2.1 STORAGE	2590
6.824.2.2 VEC_TYPE	2590
6.824.3 Constructor & Destructor Documentation	2590
6.824.3.1 WhichMin()	2590



---

6.824.4 Member Function Documentation	2590
6.824.4.1 get()	2591
6.824.5 Member Data Documentation	2591
6.824.5.1 obj	2591
6.825 Rcpp::sugar::WhichMin< RTYPE, false, T > Class Template Reference	2592
6.825.1 Detailed Description	2593
6.825.2 Member Typedef Documentation	2593
6.825.2.1 STORAGE	2593
6.825.2.2 VEC_TYPE	2593
6.825.3 Constructor & Destructor Documentation	2593
6.825.3.1 WhichMin()	2593
6.825.4 Member Function Documentation	2593
6.825.4.1 get()	2594
6.825.5 Member Data Documentation	2594
6.825.5.1 obj	2594
6.826 Rcpp::stats::WilcoxGenerator Class Reference	2595
6.826.1 Detailed Description	2596
6.826.2 Constructor & Destructor Documentation	2596
6.826.2.1 WilcoxGenerator()	2596
6.826.3 Member Function Documentation	2596
6.826.3.1 operator>()	2596
6.826.4 Member Data Documentation	2596
6.826.4.1 mm	2596
6.826.4.2 nn	2597
6.827 Rcpp::traits::wrap_type_char_array Struct Reference	2597
6.827.1 Detailed Description	2597
6.828 Rcpp::traits::wrap_type_enum_tag Struct Reference	2597
6.828.1 Detailed Description	2597
6.829 Rcpp::traits::wrap_type_module_object_pointer_tag Struct Reference	2598
6.829.1 Detailed Description	2598
6.830 Rcpp::traits::wrap_type_module_object_tag Struct Reference	2598
6.830.1 Detailed Description	2598
6.831 Rcpp::traits::wrap_type_primitive_tag Struct Reference	2598
6.831.1 Detailed Description	2598
6.832 Rcpp::traits::wrap_type_traits< T > Struct Template Reference	2599
6.832.1 Detailed Description	2599
6.832.2 Member Typedef Documentation	2599
6.832.2.1 wrap_category	2599
6.833 Rcpp::traits::wrap_type_traits< bool > Struct Reference	2599

---

6.833.1 Detailed Description	2600
6.833.2 Member Typedef Documentation	2600
6.833.2.1 wrap_category	2600
6.834 Rcpp::traits::wrap_type_traits< char > Struct Reference	2600
6.834.1 Detailed Description	2600
6.834.2 Member Typedef Documentation	2600
6.834.2.1 wrap_category	2601
6.835 Rcpp::traits::wrap_type_traits< char[N]> Struct Template Reference	2601
6.835.1 Detailed Description	2601
6.835.2 Member Typedef Documentation	2601
6.835.2.1 wrap_category	2601
6.836 Rcpp::traits::wrap_type_traits< const char[N]> Struct Template Reference	2602
6.836.1 Detailed Description	2602
6.836.2 Member Typedef Documentation	2602
6.836.2.1 wrap_category	2602
6.837 Rcpp::traits::wrap_type_traits< const int > Struct Reference	2602
6.837.1 Detailed Description	2603
6.837.2 Member Typedef Documentation	2603
6.837.2.1 wrap_category	2603
6.838 Rcpp::traits::wrap_type_traits< double > Struct Reference	2603
6.838.1 Detailed Description	2603
6.838.2 Member Typedef Documentation	2603
6.838.2.1 wrap_category	2603
6.839 Rcpp::traits::wrap_type_traits< float > Struct Reference	2604
6.839.1 Detailed Description	2604
6.839.2 Member Typedef Documentation	2604
6.839.2.1 wrap_category	2604
6.840 Rcpp::traits::wrap_type_traits< int > Struct Reference	2604
6.840.1 Detailed Description	2604
6.840.2 Member Typedef Documentation	2605
6.840.2.1 wrap_category	2605
6.841 Rcpp::traits::wrap_type_traits< long > Struct Reference	2605
6.841.1 Detailed Description	2605
6.841.2 Member Typedef Documentation	2605
6.841.2.1 wrap_category	2605
6.842 Rcpp::traits::wrap_type_traits< long double > Struct Reference	2606
6.842.1 Detailed Description	2606
6.842.2 Member Typedef Documentation	2606
6.842.2.1 wrap_category	2606

---

6.843 Rcpp::traits::wrap_type_traits< Rbyte > Struct Reference	2606
6.843.1 Detailed Description	2606
6.843.2 Member Typedef Documentation	2607
6.843.2.1 wrap_category	2607
6.844 Rcpp::traits::wrap_type_traits< Rcomplex > Struct Reference	2607
6.844.1 Detailed Description	2607
6.844.2 Member Typedef Documentation	2607
6.844.2.1 wrap_category	2607
6.845 Rcpp::traits::wrap_type_traits< Rcpp::Date > Struct Reference	2608
6.845.1 Detailed Description	2608
6.845.2 Member Typedef Documentation	2608
6.845.2.1 wrap_category	2608
6.846 Rcpp::traits::wrap_type_traits< Rcpp::Datetime > Struct Reference	2608
6.846.1 Detailed Description	2608
6.846.2 Member Typedef Documentation	2609
6.846.2.1 wrap_category	2609
6.847 Rcpp::traits::wrap_type_traits< Rcpp::object< T > > Struct Template Reference	2609
6.847.1 Detailed Description	2609
6.847.2 Member Typedef Documentation	2609
6.847.2.1 wrap_category	2609
6.848 Rcpp::traits::wrap_type_traits< Rcpp::String > Struct Reference	2610
6.848.1 Detailed Description	2610
6.848.2 Member Typedef Documentation	2610
6.848.2.1 wrap_category	2610
6.849 Rcpp::traits::wrap_type_traits< short > Struct Reference	2610
6.849.1 Detailed Description	2610
6.849.2 Member Typedef Documentation	2611
6.849.2.1 wrap_category	2611
6.850 Rcpp::traits::wrap_type_traits< std::complex< double > > Struct Reference	2611
6.850.1 Detailed Description	2611
6.850.2 Member Typedef Documentation	2611
6.850.2.1 wrap_category	2611
6.851 Rcpp::traits::wrap_type_traits< std::complex< float > > Struct Reference	2612
6.851.1 Detailed Description	2612
6.851.2 Member Typedef Documentation	2612
6.851.2.1 wrap_category	2612
6.852 Rcpp::traits::wrap_type_traits< std::string > Struct Reference	2612
6.852.1 Detailed Description	2612
6.852.2 Member Typedef Documentation	2613

---

6.852.2.1 wrap_category	2613
6.853 Rcpp::traits::wrap_type_traits< std::wstring > Struct Reference	2613
6.853.1 Detailed Description	2613
6.853.2 Member Typedef Documentation	2613
6.853.2.1 wrap_category	2613
6.854 Rcpp::traits::wrap_type_traits< unsigned int > Struct Reference	2614
6.854.1 Detailed Description	2614
6.854.2 Member Typedef Documentation	2614
6.854.2.1 wrap_category	2614
6.855 Rcpp::traits::wrap_type_traits< unsigned long > Struct Reference	2614
6.855.1 Detailed Description	2614
6.855.2 Member Typedef Documentation	2615
6.855.2.1 wrap_category	2615
6.856 Rcpp::traits::wrap_type_traits< unsigned short > Struct Reference	2615
6.856.1 Detailed Description	2615
6.856.2 Member Typedef Documentation	2615
6.856.2.1 wrap_category	2615
6.857 Rcpp::traits::wrap_type_traits< wchar_t > Struct Reference	2616
6.857.1 Detailed Description	2616
6.857.2 Member Typedef Documentation	2616
6.857.2.1 wrap_category	2616
6.858 Rcpp::traits::wrap_type_unknown_tag Struct Reference	2616
6.858.1 Detailed Description	2616
6.859 Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > Class Template Reference	2617
6.859.1 Detailed Description	2619
6.859.2 Member Typedef Documentation	2619
6.859.2.1 Storage	2619
6.859.2.2 unspecified_bool_type	2620
6.859.3 Constructor & Destructor Documentation	2620
6.859.3.1 XPtr() [1/4]	2620
6.859.3.2 XPtr() [2/4]	2621
6.859.3.3 XPtr() [3/4]	2622
6.859.3.4 XPtr() [4/4]	2623
6.859.4 Member Function Documentation	2623
6.859.4.1 checked_get()	2623
6.859.4.2 checked_set()	2623
6.859.4.3 get()	2624
6.859.4.4 operator T*()	2624
6.859.4.5 operator unspecified_bool_type()	2624

6.859.4.6 operator"!()	2625
6.859.4.7 operator*()	2626
6.859.4.8 operator->()	2626
6.859.4.9 operator=()	2627
6.859.4.10 release()	2627
6.859.4.11 setDeleteFinalizer()	2627
6.859.4.12 unspecified_bool_true()	2628
6.859.4.13 update()	2628
6.860 Rcpp::traits::zero_type< T > Class Template Reference	2628
6.860.1 Detailed Description	2628
6.860.2 Member Function Documentation	2629
6.860.2.1 op() [1/2]	2629
6.860.2.2 op() [2/2]	2629
6.860.2.3 operator T()	2629
<b>7 File Documentation</b>	<b>2631</b>
7.1 inst/examples/Attributes/cppFunction.R File Reference	2631
7.2 inst/examples/Attributes/Depends.cpp File Reference	2631
7.2.1 Function Documentation	2631
7.2.1.1 fastLm()	2632
7.3 inst/examples/Attributes/Export.cpp File Reference	2632
7.3.1 Function Documentation	2633
7.3.1.1 convolve()	2633
7.3.1.2 fibonacci()	2633
7.3.1.3 lapplyCpp()	2634
7.4 inst/examples/Attributes/sourceCpp.R File Reference	2634
7.5 inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp File Reference	2634
7.5.1 Function Documentation	2635
7.5.1.1 convolve10cpp()	2635
7.6 inst/examples/ConvolveBenchmarks/convolve10_cpp.h File Reference	2635
7.7 inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp File Reference	2636
7.7.1 Function Documentation	2636
7.7.1.1 convolve11cpp()	2636
7.8 inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp File Reference	2637
7.8.1 Function Documentation	2637
7.8.1.1 convolve12cpp()	2637
7.9 inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp File Reference	2638
7.9.1 Function Documentation	2638
7.9.1.1 convolve()	2638

---

7.9.1.2 convolve13cpp()	2638
7.10 inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp File Reference	2639
7.10.1 Function Documentation	2639
7.10.1.1 convolve14cpp()	2639
7.11 inst/examples/ConvolveBenchmarks/convolve2_c.c File Reference	2640
7.11.1 Function Documentation	2640
7.11.1.1 convolve2()	2641
7.12 inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp File Reference	2641
7.12.1 Function Documentation	2641
7.12.1.1 convolve3cpp()	2641
7.13 inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp File Reference	2642
7.13.1 Function Documentation	2642
7.13.1.1 convolve4cpp()	2642
7.14 inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp File Reference	2643
7.14.1 Function Documentation	2643
7.14.1.1 convolve5cpp()	2644
7.15 inst/examples/ConvolveBenchmarks/convolve7_c.c File Reference	2644
7.15.1 Function Documentation	2645
7.15.1.1 convolve7()	2645
7.16 inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp File Reference	2645
7.16.1 Function Documentation	2645
7.16.1.1 convolve8cpp()	2646
7.17 inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp File Reference	2646
7.17.1 Function Documentation	2647
7.17.1.1 convolve9cpp()	2647
7.18 inst/examples/ConvolveBenchmarks/exampleRCode.r File Reference	2647
7.19 inst/examples/ConvolveBenchmarks/loopmacro.h File Reference	2647
7.19.1 Macro Definition Documentation	2648
7.19.1.1 LOOPMACRO_C	2648
7.19.1.2 LOOPMACRO_CPP	2648
7.20 inst/examples/ConvolveBenchmarks/overhead.r File Reference	2648
7.21 inst/examples/ConvolveBenchmarks/overhead_1.cpp File Reference	2648
7.21.1 Function Documentation	2649
7.21.1.1 overhead_cpp()	2649
7.21.1.2 R_init_overhead_1()	2649
7.22 inst/examples/ConvolveBenchmarks/overhead_2.c File Reference	2650
7.22.1 Function Documentation	2650
7.22.1.1 overhead_c()	2650
7.22.1.2 R_init_overhead_2()	2651

---

7.23	inst/examples/FastLM/benchmark.r File Reference	2651
7.24	inst/examples/FastLM/benchmarkLongley.r File Reference	2651
7.25	inst/examples/FastLM/fastLMviaArmadillo.r File Reference	2651
7.26	inst/examples/FastLM/fastLMviaGSL.r File Reference	2651
7.27	inst/examples/FastLM/lmArmadillo.R File Reference	2651
7.28	inst/examples/FastLM/lmGSL.R File Reference	2651
7.29	inst/examples/functionCallback/newApiExample.r File Reference	2651
7.30	inst/examples/Misc/fibonacci.r File Reference	2651
7.31	inst/examples/Misc/iffelseLooped.r File Reference	2651
7.32	inst/examples/Misc/newFib.r File Reference	2651
7.33	inst/examples/Misc/piBySimulation.r File Reference	2651
7.34	inst/examples/Misc/piSugar.cpp File Reference	2651
7.34.1	Function Documentation	2652
7.34.1.1	piSugar()	2652
7.35	inst/examples/OpenMP/check.R File Reference	2653
7.36	inst/examples/OpenMP/OpenMPandInline.r File Reference	2653
7.37	inst/examples/OpenMP/piWithInterrupts.cpp File Reference	2653
7.37.1	Function Documentation	2653
7.37.1.1	check_interrupt()	2653
7.37.1.2	check_interrupt_impl()	2654
7.37.1.3	PiLeibniz()	2655
7.38	inst/examples/performance/extractors.R File Reference	2656
7.39	inst/examples/performance/performance.R File Reference	2656
7.40	inst/examples/RcppGibbs/RcppGibbs.R File Reference	2656
7.41	inst/examples/RcppGibbs/RcppGibbs_Updated.R File Reference	2656
7.42	inst/examples/RcppGibbs/timeRNGs.R File Reference	2656
7.43	inst/examples/RcppInline/external_pointer.r File Reference	2656
7.44	inst/examples/RcppInline/RcppInlineExample.r File Reference	2656
7.45	inst/examples/RcppInline/RcppInlineWithLibsExamples.r File Reference	2656
7.46	inst/examples/RcppInline/RcppSimpleExample.r File Reference	2656
7.47	inst/examples/RcppInline/RObject.r File Reference	2656
7.48	inst/examples/RcppInline/UncaughtExceptions.r File Reference	2656
7.49	inst/examples/SugarPerformance/sugarBenchmarks.R File Reference	2656
7.50	inst/examples/SugarPerformance/Timertest.cpp File Reference	2656
7.50.1	Function Documentation	2657
7.50.1.1	main()	2657
7.51	inst/include/doxygen/Examples.h File Reference	2658
7.52	inst/include/Rcpp.h File Reference	2658
7.52.1	Macro Definition Documentation	2659

---

7.52.1.1 RCPP_NEW_DATE_DATETIME_VECTORS . . . . .	2659
7.53 inst/include/Rcpp/algo.h File Reference . . . . .	2660
7.54 inst/include/Rcpp/algorithm.h File Reference . . . . .	2661
7.54.1 Macro Definition Documentation . . . . .	2663
7.54.1.1 RCPP_CONSTEXPR_FUNC . . . . .	2663
7.54.1.2 RCPP_CONSTEXPR_VAR . . . . .	2663
7.55 inst/include/Rcpp/api/bones/bones.h File Reference . . . . .	2663
7.56 inst/include/Rcpp/api/bones/Date.h File Reference . . . . .	2664
7.57 inst/include/Rcpp/api/meat/Date.h File Reference . . . . .	2665
7.58 inst/include/Rcpp/date_datetime/Date.h File Reference . . . . .	2665
7.59 inst/include/Rcpp/api/bones/Datetime.h File Reference . . . . .	2666
7.60 inst/include/Rcpp/api/meat/Datetime.h File Reference . . . . .	2667
7.61 inst/include/Rcpp/date_datetime/Datetime.h File Reference . . . . .	2667
7.62 inst/include/Rcpp/api/bones/wrap_extra_steps.h File Reference . . . . .	2668
7.63 inst/include/Rcpp/api/meat/as.h File Reference . . . . .	2669
7.64 inst/include/Rcpp/as.h File Reference . . . . .	2669
7.65 inst/include/Rcpp/api/meat/DataFrame.h File Reference . . . . .	2671
7.66 inst/include/Rcpp/DataFrame.h File Reference . . . . .	2671
7.67 inst/include/Rcpp/api/meat/Dimension.h File Reference . . . . .	2672
7.68 inst/include/Rcpp/Dimension.h File Reference . . . . .	2672
7.69 inst/include/Rcpp/api/meat/DottedPairImpl.h File Reference . . . . .	2673
7.70 inst/include/Rcpp/DottedPairImpl.h File Reference . . . . .	2673
7.71 inst/include/Rcpp/api/meat/Environment.h File Reference . . . . .	2673
7.72 inst/include/Rcpp/Environment.h File Reference . . . . .	2674
7.73 inst/include/Rcpp/api/meat/export.h File Reference . . . . .	2675
7.74 inst/include/Rcpp/api/meat/is.h File Reference . . . . .	2676
7.75 inst/include/Rcpp/is.h File Reference . . . . .	2677
7.76 inst/include/Rcpp/sugar/logical/is.h File Reference . . . . .	2678
7.77 inst/include/Rcpp/api/meat/meat.h File Reference . . . . .	2678
7.78 inst/include/Rcpp/api/meat/message.h File Reference . . . . .	2679
7.79 inst/include/Rcpp/api/meat/module/Module.h File Reference . . . . .	2679
7.80 inst/include/Rcpp/module/Module.h File Reference . . . . .	2680
7.81 inst/include/Rcpp/Module.h File Reference . . . . .	2680
7.81.1 Macro Definition Documentation . . . . .	2682
7.81.1.1 GET_MODULE_SYM . . . . .	2682
7.81.1.2 LOAD_RCPP_MODULE . . . . .	2682
7.81.1.3 RCPP_MODULE . . . . .	2682
7.81.1.4 RCPP_MODULE_BOOT . . . . .	2683
7.81.1.5 VARIABLE_IS_NOT_USED . . . . .	2683



---

7.81.2 Variable Documentation	2683
7.81.2.1 moduleSym	2683
7.82 inst/include/Rcpp/api/meat/protection.h File Reference	2683
7.83 inst/include/Rcpp/protection/protection.h File Reference	2684
7.84 inst/include/Rcpp/api/meat/proxy.h File Reference	2684
7.85 inst/include/Rcpp/proxy/proxy.h File Reference	2685
7.86 inst/include/Rcpp/vector/proxy.h File Reference	2685
7.87 inst/include/Rcpp/api/meat/Rcpp_eval.h File Reference	2686
7.88 inst/include/Rcpp/api/meat/S4.h File Reference	2687
7.89 inst/include/Rcpp/S4.h File Reference	2687
7.90 inst/include/Rcpp/api/meat/StretchyList.h File Reference	2688
7.91 inst/include/Rcpp/StretchyList.h File Reference	2688
7.92 inst/include/Rcpp/api/meat/Vector.h File Reference	2689
7.93 inst/include/Rcpp/vector/Vector.h File Reference	2690
7.94 inst/include/Rcpp/Vector.h File Reference	2691
7.95 inst/include/Rcpp/api/meat/wrap.h File Reference	2692
7.96 inst/include/Rcpp/barrier.h File Reference	2692
7.96.1 Function Documentation	2693
7.96.1.1 char_get_string_elt()	2693
7.96.1.2 char_nocheck()	2693
7.96.1.3 char_set_string_elt()	2694
7.96.1.4 dataptr()	2694
7.96.1.5 get_string_elt()	2694
7.96.1.6 get_string_ptr()	2694
7.96.1.7 get_vector_elt()	2695
7.96.1.8 get_vector_ptr()	2695
7.96.1.9 set_string_elt()	2695
7.96.1.10 set_vector_elt()	2695
7.97 inst/include/Rcpp/Benchmark/Timer.h File Reference	2696
7.97.1 Macro Definition Documentation	2696
7.97.1.1 R_NO_REMAP	2697
7.98 inst/examples/SugarPerformance/Timer.h File Reference	2697
7.99 inst/include/Rcpp/clone.h File Reference	2697
7.100 inst/include/Rcpp/complex.h File Reference	2698
7.100.1 Macro Definition Documentation	2698
7.100.1.1 dplyr_tools_complex_H	2698
7.100.2 Function Documentation	2699
7.100.2.1 operator*()	2699
7.100.2.2 operator+()	2699

---

7.100.2.3 operator-()	2699
7.100.2.4 operator/()	2699
7.100.2.5 operator<<()	2700
7.100.2.6 operator==( )	2700
7.101 inst/include/Rcpp/sugar/functions/complex.h File Reference	2700
7.101.1 Macro Definition Documentation	2701
7.101.1.1 RCPP_SUGAR_COMPLEX	2701
7.102 inst/include/Rcpp/config.h File Reference	2701
7.102.1 Macro Definition Documentation	2702
7.102.1.1 RCPP_DEV_VERSION	2702
7.102.1.2 RCPP_DEV_VERSION_STRING	2702
7.102.1.3 Rcpp_Version	2702
7.102.1.4 RCPP_VERSION	2702
7.102.1.5 RCPP_VERSION_STRING	2703
7.102.1.6 RcppDevVersion	2703
7.103 inst/include/Rcpp/date_datetime/date_datetime.h File Reference	2703
7.104 inst/include/Rcpp/date_datetime/newDatetimeVector.h File Reference	2704
7.105 inst/include/Rcpp/date_datetime/newDateVector.h File Reference	2704
7.106 inst/include/Rcpp/date_datetime/oldDatetimeVector.h File Reference	2705
7.107 inst/include/Rcpp/date_datetime/oldDateVector.h File Reference	2706
7.108 inst/include/Rcpp/DottedPair.h File Reference	2706
7.109 inst/include/Rcpp/exceptions/cpp11/exceptions.h File Reference	2707
7.109.1 Macro Definition Documentation	2708
7.109.1.1 RCPP_ADVANCED_EXCEPTION_CLASS	2708
7.110 inst/include/Rcpp/exceptions/cpp98/exceptions.h File Reference	2708
7.110.1 Macro Definition Documentation	2710
7.110.1.1 RCPP_ADVANCED_EXCEPTION_CLASS	2710
7.111 inst/include/Rcpp/exceptions.h File Reference	2710
7.111.1 Macro Definition Documentation	2712
7.111.1.1 DEMANGLE	2712
7.111.1.2 GET_STACKTRACE	2712
7.111.1.3 RCPP_DEFAULT_INCLUDE_CALL	2712
7.111.1.4 RCPP_EXCEPTION_CLASS	2713
7.111.1.5 RCPP_SIMPLE_EXCEPTION_CLASS	2713
7.111.2 Function Documentation	2713
7.111.2.1 demangle()	2713
7.111.2.2 exception_to_condition_template()	2714
7.111.2.3 exception_to_r_condition()	2714
7.111.2.4 exception_to_try_error()	2715

---

7.111.2.5 forward_exception_to_r()	2715
7.111.2.6 forward_rcpp_exception_to_r()	2716
7.111.2.7 get_exception_classes()	2716
7.111.2.8 get_last_call()	2716
7.111.2.9 make_condition()	2717
7.111.2.10 rcpp_exception_to_r_condition()	2717
7.111.2.11 string_to_try_error()	2717
7.112 inst/include/Rcpp/exceptions_impl.h File Reference	2718
7.112.1 Macro Definition Documentation	2718
7.112.1.1 RCPP_DEMANGLER_ENABLED	2718
7.113 inst/include/Rcpp/Extractor.h File Reference	2718
7.114 inst/include/Rcpp/Fast.h File Reference	2719
7.115 inst/include/Rcpp/Formula.h File Reference	2719
7.116 inst/include/Rcpp/Function.h File Reference	2720
7.117 inst/include/Rcpp/grow.h File Reference	2721
7.118 inst/include/Rcpp/hash/hash.h File Reference	2722
7.119 inst/include/Rcpp/hash/IndexHash.h File Reference	2722
7.119.1 Macro Definition Documentation	2723
7.119.1.1 RCPP_HASH	2723
7.119.1.2 RCPP_PROFILE_RECORD	2723
7.119.1.3 RCPP_PROFILE_TIC	2723
7.119.1.4 RCPP_PROFILE_TOC	2723
7.119.1.5 RCPP_USE_CACHE_HASH	2724
7.120 inst/include/Rcpp/hash/SelfHash.h File Reference	2724
7.121 inst/include/Rcpp/InputParameter.h File Reference	2724
7.122 inst/include/Rcpp/InternalFunction.h File Reference	2725
7.123 inst/include/Rcpp/InternalFunctionWithStdFunction.h File Reference	2726
7.124 inst/include/Rcpp/Interrupt.h File Reference	2726
7.125 inst/include/Rcpp/iostream/Rstreambuf.h File Reference	2728
7.126 inst/include/Rcpp/lang.h File Reference	2729
7.126.1 Macro Definition Documentation	2730
7.126.1.1 Rcpp_lang1	2730
7.126.1.2 Rcpp_lang2	2731
7.126.1.3 Rcpp_lang3	2731
7.126.1.4 Rcpp_lang4	2731
7.126.1.5 Rcpp_lang5	2731
7.126.1.6 Rcpp_lang6	2731
7.126.1.7 Rcpp_lcons	2731
7.126.1.8 Rcpp_list1	2732

---

7.127 inst/include/Rcpp/Language.h File Reference	2732
7.128 inst/include/Rcpp/longlong.h File Reference	2733
7.129 inst/include/Rcpp/traits/longlong.h File Reference	2733
7.130 inst/include/Rcpp/macros/cat.hpp File Reference	2734
7.130.1 Macro Definition Documentation	2734
7.130.1.1 RCPP_PP_CAT	2734
7.130.1.2 RCPP_PP_CAT_I	2734
7.130.1.3 RCPP_PP_CAT_II	2735
7.130.1.4 RCPP_PP_CAT_OO	2735
7.131 inst/include/Rcpp/macros/config.hpp File Reference	2735
7.131.1 Macro Definition Documentation	2735
7.131.1.1 RCPP_PP_CONFIG_BCC	2736
7.131.1.2 RCPP_PP_CONFIG_DMC	2736
7.131.1.3 RCPP_PP_CONFIG_EDG	2736
7.131.1.4 RCPP_PP_CONFIG_ERRORS	2736
7.131.1.5 RCPP_PP_CONFIG_EXTENDED_LINE_INFO	2736
7.131.1.6 RCPP_PP_CONFIG_FLAGS	2736
7.131.1.7 RCPP_PP_CONFIG_IDEAL	2737
7.131.1.8 RCPP_PP_CONFIG_MSVC	2737
7.131.1.9 RCPP_PP_CONFIG_MWCC	2737
7.131.1.10 RCPP_PP_CONFIG_STRICT	2737
7.132 inst/include/Rcpp/macros/debug.h File Reference	2737
7.132.1 Macro Definition Documentation	2738
7.132.1.1 RCPP_DEBUG	2738
7.132.1.2 RCPP_DEBUG_1	2738
7.132.1.3 RCPP_DEBUG_2	2738
7.132.1.4 RCPP_DEBUG_3	2739
7.132.1.5 RCPP_DEBUG_4	2739
7.132.1.6 RCPP_DEBUG_5	2739
7.132.1.7 RCPP_DEBUG_LEVEL	2739
7.132.1.8 RCPP_DEBUG_MODULE	2740
7.132.1.9 RCPP_DEBUG_MODULE_1	2740
7.132.1.10 RCPP_DEBUG_MODULE_2	2740
7.132.1.11 RCPP_DEBUG_MODULE_3	2740
7.132.1.12 RCPP_DEBUG_MODULE_4	2741
7.132.1.13 RCPP_DEBUG_MODULE_5	2741
7.132.1.14 RCPP_DEBUG_MODULE_LEVEL	2741
7.133 inst/include/Rcpp/macros/dispatch.h File Reference	2741
7.133.1 Macro Definition Documentation	2742

---

7.133.1.1	__RCPP_HANDLE_CASE__	2742
7.133.1.2	__RCPP_RETURN__	2742
7.133.1.3	RCPP_RETURN_MATRIX	2743
7.133.1.4	RCPP_RETURN_VECTOR	2743
7.134	inst/include/Rcpp/macros/interface.h File Reference	2743
7.134.1	Macro Definition Documentation	2743
7.134.1.1	RCPP_API_CLASS	2744
7.134.1.2	RCPP_CTOR_ASSIGN	2744
7.134.1.3	RCPP_CTOR_ASSIGN_WITH_BASE	2744
7.134.1.4	RCPP_GENERATE_CTOR_ASSIGN	2745
7.135	inst/include/Rcpp/macros/macros.h File Reference	2745
7.135.1	Macro Definition Documentation	2746
7.135.1.1	BEGIN_RCPP	2746
7.135.1.2	END_RCPP	2746
7.135.1.3	END_RCPP_RETURN_ERROR	2747
7.135.1.4	RCPP_DECORATE	2747
7.135.1.5	Rcpp_error	2747
7.135.1.6	RCPP_GET_CLASS	2747
7.135.1.7	RCPP_GET_NAMES	2748
7.135.1.8	VOID_END_RCPP	2748
7.136	inst/include/Rcpp/stats/dpq/macros.h File Reference	2748
7.136.1	Macro Definition Documentation	2749
7.136.1.1	give_log	2749
7.136.1.2	R_D_0	2749
7.136.1.3	R_D_1	2749
7.136.1.4	R_D_Clog	2750
7.136.1.5	R_D_Cval	2750
7.136.1.6	R_D_exp	2750
7.136.1.7	R_D_fexp	2750
7.136.1.8	R_D_forceint	2750
7.136.1.9	R_D_LExp	2751
7.136.1.10	R_D_log	2751
7.136.1.11	R_D_Lval	2751
7.136.1.12	R_D_neglNonint	2751
7.136.1.13	R_D_nonint	2751
7.136.1.14	R_D_nonint_check	2752
7.136.1.15	R_D_qlv	2752
7.136.1.16	R_D_val	2752
7.136.1.17	R_DT_0	2752

---

7.136.1.18 R_DT_1	2752
7.136.1.19 R_DT_Cexp	2753
7.136.1.20 R_DT_Clv	2753
7.136.1.21 R_DT_Clog	2753
7.136.1.22 R_DT_Cval	2753
7.136.1.23 R_DT_exp	2753
7.136.1.24 R_DT_log	2754
7.136.1.25 R_DT_Log	2754
7.136.1.26 R_DT_qlv	2754
7.136.1.27 R_DT_val	2754
7.136.1.28 R_Log1_Exp	2754
7.136.1.29 R_P_bounds_01	2755
7.136.1.30 R_P_bounds_Inf_01	2755
7.136.1.31 R_Q_P01_boundaries	2755
7.136.1.32 R_Q_P01_check	2756
7.137 inst/include/Rcpp/macros/module.h File Reference	2756
7.137.1 Macro Definition Documentation	2756
7.137.1.1 RCPP_EXPOSED_AS	2756
7.137.1.2 RCPP_EXPOSED_CLASS	2757
7.137.1.3 RCPP_EXPOSED_CLASS_NODECL	2757
7.137.1.4 RCPP_EXPOSED_ENUM	2757
7.137.1.5 RCPP_EXPOSED_ENUM_AS	2757
7.137.1.6 RCPP_EXPOSED_ENUM_NODECL	2758
7.137.1.7 RCPP_EXPOSED_ENUM_WRAP	2758
7.137.1.8 RCPP_EXPOSED_WRAP	2758
7.138 inst/include/Rcpp/macros/traits.h File Reference	2758
7.138.1 Macro Definition Documentation	2759
7.138.1.1 RCPP_ENUM_TRAITS	2759
7.138.1.2 RCPP_TRAITS	2759
7.139 inst/include/Rcpp/traits/traits.h File Reference	2759
7.140 inst/include/Rcpp/vector/traits.h File Reference	2761
7.141 inst/include/Rcpp/macros/unroll.h File Reference	2761
7.141.1 Macro Definition Documentation	2761
7.141.1.1 RCPP_LOOP_UNROLL	2762
7.141.1.2 RCPP_LOOP_UNROLL_LHSFUN	2762
7.141.1.3 RCPP_LOOP_UNROLL_PTR	2763
7.142 inst/include/Rcpp/macros/xp.h File Reference	2763
7.142.1 Macro Definition Documentation	2763
7.142.1.1 RCPP_XP_FIELD	2764

---

7.142.1.2 RCPP_XP_FIELD_GET	2764
7.142.1.3 RCPP_XP_FIELD_SET	2764
7.143 inst/include/Rcpp/module/class.h File Reference	2765
7.144 inst/include/Rcpp/module/class_Base.h File Reference	2765
7.145 inst/include/Rcpp/module/CppFunction.h File Reference	2766
7.146 inst/include/Rcpp/module/get_return_type.h File Reference	2766
7.147 inst/include/Rcpp/module/Module_Add_Property.h File Reference	2767
7.147.1 Function Documentation	2767
7.147.1.1 property() [1/6]	2768
7.147.1.2 property() [2/6]	2768
7.147.1.3 property() [3/6]	2768
7.147.1.4 property() [4/6]	2768
7.147.1.5 property() [5/6]	2769
7.147.1.6 property() [6/6]	2769
7.148 inst/include/Rcpp/module/Module_Field.h File Reference	2769
7.148.1 Function Documentation	2770
7.148.1.1 field()	2770
7.148.1.2 field_readonly()	2770
7.149 inst/include/Rcpp/module/Module_Property.h File Reference	2770
7.150 inst/include/Rcpp/Na_Proxy.h File Reference	2771
7.151 inst/include/Rcpp/Named.h File Reference	2771
7.152 inst/include/Rcpp/Nullable.h File Reference	2772
7.153 inst/include/Rcpp/Pairlist.h File Reference	2772
7.154 inst/include/Rcpp/platform/compiler.h File Reference	2773
7.154.1 Macro Definition Documentation	2774
7.154.1.1 RCPP_UNORDERED_MAP	2774
7.154.1.2 RCPP_UNORDERED_SET	2774
7.154.1.3 RCPP_USING_MAP	2774
7.154.1.4 RCPP_USING_SET	2774
7.155 inst/include/Rcpp/platform/solaris.h File Reference	2775
7.156 inst/include/Rcpp/print.h File Reference	2775
7.157 inst/include/Rcpp/Promise.h File Reference	2775
7.158 inst/include/Rcpp/protection/Armor.h File Reference	2776
7.159 inst/include/Rcpp/protection/Shelter.h File Reference	2776
7.160 inst/include/Rcpp/protection/Shield.h File Reference	2777
7.161 inst/include/Rcpp/proxy/AttributeProxy.h File Reference	2777
7.162 inst/include/Rcpp/proxy/Binding.h File Reference	2778
7.163 inst/include/Rcpp/proxy/DottedPairProxy.h File Reference	2778
7.164 inst/include/Rcpp/proxy/FieldProxy.h File Reference	2779

---

7.165 inst/include/Rcpp/proxy/GenericProxy.h File Reference . . . . .	2779
7.166 inst/include/Rcpp/proxy/NamesProxy.h File Reference . . . . .	2780
7.167 inst/include/Rcpp/proxy/ProtectedProxy.h File Reference . . . . .	2780
7.168 inst/include/Rcpp/proxy/RObjectMethods.h File Reference . . . . .	2781
7.169 inst/include/Rcpp/proxy/SlotProxy.h File Reference . . . . .	2781
7.170 inst/include/Rcpp/proxy/TagProxy.h File Reference . . . . .	2782
7.171 inst/include/Rcpp/r/headers.h File Reference . . . . .	2782
7.171.1 Macro Definition Documentation . . . . .	2783
7.171.1.1 MAXELTSIZE . . . . .	2783
7.171.1.2 NORET . . . . .	2783
7.171.1.3 R_NO_REMAP . . . . .	2784
7.171.1.4 STRICT_R_HEADERS . . . . .	2784
7.172 inst/include/Rcpp/r_cast.h File Reference . . . . .	2784
7.173 inst/include/Rcpp/Reference.h File Reference . . . . .	2785
7.174 inst/include/Rcpp/Rmath.h File Reference . . . . .	2786
7.175 inst/include/Rcpp/RNGScope.h File Reference . . . . .	2789
7.176 inst/include/Rcpp/RObject.h File Reference . . . . .	2789
7.177 inst/include/Rcpp/routines.h File Reference . . . . .	2790
7.177.1 Macro Definition Documentation . . . . .	2791
7.177.1.1 GET_CALLABLE . . . . .	2792
7.177.2 Function Documentation . . . . .	2792
7.177.2.1 char_get_string_elt() . . . . .	2792
7.177.2.2 char_nocheck() . . . . .	2792
7.177.2.3 char_set_string_elt() . . . . .	2792
7.177.2.4 dataptr() . . . . .	2793
7.177.2.5 demangle() . . . . .	2793
7.177.2.6 error_occured() . . . . .	2793
7.177.2.7 get_cache() . . . . .	2793
7.177.2.8 get_string_elt() . . . . .	2794
7.177.2.9 get_string_ptr() . . . . .	2794
7.177.2.10 get_vector_elt() . . . . .	2794
7.177.2.11 get_vector_ptr() . . . . .	2794
7.177.2.12 getCurrentScope() . . . . .	2794
7.177.2.13 rcpp_get_current_error() . . . . .	2795
7.177.2.14 rcpp_get_stack_trace() . . . . .	2795
7.177.2.15 rcpp_set_stack_trace() . . . . .	2795
7.177.2.16 reset_current_error() . . . . .	2795
7.177.2.17 set_string_elt() . . . . .	2796
7.177.2.18 set_vector_elt() . . . . .	2796



---

7.177.2.19 <code>setCurrentScope()</code> . . . . .	2796
7.177.2.20 <code>short_file_name()</code> . . . . .	2796
7.177.2.21 <code>stack_trace()</code> . . . . .	2797
7.178 <code>inst/include/Rcpp/sprintf.h</code> File Reference . . . . .	2797
7.179 <code>inst/include/Rcpp/stats/beta.h</code> File Reference . . . . .	2798
7.180 <code>inst/include/Rcpp/stats/binom.h</code> File Reference . . . . .	2798
7.181 <code>inst/include/Rcpp/stats/cauchy.h</code> File Reference . . . . .	2798
7.182 <code>inst/include/Rcpp/stats/chisq.h</code> File Reference . . . . .	2799
7.183 <code>inst/include/Rcpp/stats/dpq/dpq.h</code> File Reference . . . . .	2799
7.183.1 Macro Definition Documentation . . . . .	2800
7.183.1.1 <code>Rcpp_DPQ_0</code> . . . . .	2800
7.183.1.2 <code>Rcpp_DPQ_1</code> . . . . .	2801
7.183.1.3 <code>Rcpp_DPQ_2</code> . . . . .	2801
7.183.1.4 <code>Rcpp_DPQ_3</code> . . . . .	2802
7.184 <code>inst/include/Rcpp/stats/exp.h</code> File Reference . . . . .	2802
7.185 <code>inst/include/Rcpp/stats/f.h</code> File Reference . . . . .	2803
7.186 <code>inst/include/Rcpp/stats/gamma.h</code> File Reference . . . . .	2803
7.187 <code>inst/include/Rcpp/stats/geom.h</code> File Reference . . . . .	2804
7.188 <code>inst/include/Rcpp/stats/hyper.h</code> File Reference . . . . .	2804
7.189 <code>inst/include/Rcpp/stats/lnorm.h</code> File Reference . . . . .	2804
7.190 <code>inst/include/Rcpp/stats/logis.h</code> File Reference . . . . .	2805
7.191 <code>inst/include/Rcpp/stats/nbeta.h</code> File Reference . . . . .	2805
7.192 <code>inst/include/Rcpp/stats/nbinom.h</code> File Reference . . . . .	2805
7.193 <code>inst/include/Rcpp/stats/nbinom_mu.h</code> File Reference . . . . .	2806
7.194 <code>inst/include/Rcpp/stats/nchisq.h</code> File Reference . . . . .	2806
7.195 <code>inst/include/Rcpp/stats/nf.h</code> File Reference . . . . .	2806
7.196 <code>inst/include/Rcpp/stats/norm.h</code> File Reference . . . . .	2806
7.197 <code>inst/include/Rcpp/stats/nt.h</code> File Reference . . . . .	2807
7.198 <code>inst/include/Rcpp/stats/pois.h</code> File Reference . . . . .	2807
7.199 <code>inst/include/Rcpp/stats/random/random.h</code> File Reference . . . . .	2807
7.200 <code>inst/include/Rcpp/stats/random/rbeta.h</code> File Reference . . . . .	2809
7.201 <code>inst/include/Rcpp/stats/random/rbinom.h</code> File Reference . . . . .	2809
7.202 <code>inst/include/Rcpp/stats/random/rcauchy.h</code> File Reference . . . . .	2810
7.203 <code>inst/include/Rcpp/stats/random/rchisq.h</code> File Reference . . . . .	2810
7.204 <code>inst/include/Rcpp/stats/random/rexp.h</code> File Reference . . . . .	2811
7.205 <code>inst/include/Rcpp/stats/random/rf.h</code> File Reference . . . . .	2811
7.206 <code>inst/include/Rcpp/stats/random/rgamma.h</code> File Reference . . . . .	2812
7.207 <code>inst/include/Rcpp/stats/random/rgeom.h</code> File Reference . . . . .	2812
7.208 <code>inst/include/Rcpp/stats/random/rhyper.h</code> File Reference . . . . .	2813

---

7.209	inst/include/Rcpp/stats/random/rlnorm.h File Reference	2813
7.210	inst/include/Rcpp/stats/random/rlogis.h File Reference	2814
7.211	inst/include/Rcpp/stats/random/rnbinom.h File Reference	2814
7.212	inst/include/Rcpp/stats/random/rnbinom_mu.h File Reference	2815
7.213	inst/include/Rcpp/stats/random/rnchisq.h File Reference	2815
7.214	inst/include/Rcpp/stats/random/rnorm.h File Reference	2816
7.215	inst/include/Rcpp/stats/random/rpois.h File Reference	2816
7.216	inst/include/Rcpp/stats/random/rsignrank.h File Reference	2817
7.217	inst/include/Rcpp/stats/random/rt.h File Reference	2817
7.218	inst/include/Rcpp/stats/random/runif.h File Reference	2818
7.219	inst/include/Rcpp/stats/random/rweibull.h File Reference	2818
7.220	inst/include/Rcpp/stats/random/rwilcox.h File Reference	2819
7.221	inst/include/Rcpp/stats/stats.h File Reference	2819
7.221.1	Macro Definition Documentation	2820
7.221.1.1	ML_NAN	2820
7.221.1.2	ML_NEGINF	2820
7.221.1.3	ML_POSINF	2820
7.222	inst/include/Rcpp/stats/t.h File Reference	2821
7.223	inst/include/Rcpp/stats/unif.h File Reference	2821
7.224	inst/include/Rcpp/stats/weibull.h File Reference	2821
7.225	inst/include/Rcpp/storage/NoProtectStorage.h File Reference	2822
7.226	inst/include/Rcpp/storage/PreserveStorage.h File Reference	2822
7.227	inst/include/Rcpp/storage/storage.h File Reference	2823
7.228	inst/include/Rcpp/String.h File Reference	2823
7.228.1	Macro Definition Documentation	2824
7.228.1.1	RCPP_STRING_DEBUG	2824
7.228.1.2	RCPP_STRING_DEBUG_1	2825
7.228.1.3	RCPP_STRING_DEBUG_2	2825
7.228.1.4	RCPP_STRING_DEBUG_3	2825
7.228.1.5	RCPP_STRING_DEBUG_LEVEL	2825
7.229	inst/include/Rcpp/StringTransformer.h File Reference	2825
7.230	inst/include/Rcpp/sugar/block/block.h File Reference	2826
7.231	inst/include/Rcpp/sugar/block/SugarBlock_1.h File Reference	2826
7.231.1	Macro Definition Documentation	2827
7.231.1.1	SB1_T	2827
7.231.1.2	SUGAR_BLOCK_1	2827
7.232	inst/include/Rcpp/sugar/block/SugarBlock_2.h File Reference	2828
7.232.1	Macro Definition Documentation	2828
7.232.1.1	SB2_LHT	2828

---

7.232.1.2 SB2_RHT	2829
7.232.1.3 SUGAR_BLOCK_2	2829
7.232.1.4 SUGAR_BLOCK_2_NA	2829
7.233 inst/include/Rcpp/sugar/block/SugarBlock_3.h File Reference	2829
7.233.1 Macro Definition Documentation	2830
7.233.1.1 SB3_T1	2830
7.233.1.2 SB3_T2	2830
7.233.1.3 SB3_T3	2830
7.233.1.4 SUGAR_BLOCK_3	2831
7.234 inst/include/Rcpp/sugar/block/SugarMath.h File Reference	2831
7.234.1 Macro Definition Documentation	2832
7.234.1.1 SUGAR_MATH_1	2832
7.235 inst/include/Rcpp/sugar/block/Vectorized_Math.h File Reference	2832
7.235.1 Macro Definition Documentation	2833
7.235.1.1 VECTORIZED_MATH_1	2833
7.236 inst/include/Rcpp/sugar/functions/all.h File Reference	2833
7.237 inst/include/Rcpp/sugar/functions/any.h File Reference	2834
7.238 inst/include/Rcpp/sugar/functions/cbind.h File Reference	2835
7.238.1 Macro Definition Documentation	2844
7.238.1.1 MakeBindable	2844
7.239 inst/include/Rcpp/sugar/functions/clamp.h File Reference	2844
7.240 inst/include/Rcpp/sugar/functions/cummax.h File Reference	2845
7.241 inst/include/Rcpp/sugar/functions/cummin.h File Reference	2845
7.242 inst/include/Rcpp/sugar/functions/cumprod.h File Reference	2846
7.243 inst/include/Rcpp/sugar/functions/cumsum.h File Reference	2846
7.244 inst/include/Rcpp/sugar/functions/diff.h File Reference	2847
7.245 inst/include/Rcpp/sugar/functions/duplicated.h File Reference	2848
7.246 inst/include/Rcpp/sugar/functions/functions.h File Reference	2848
7.247 inst/include/Rcpp/sugar/functions/head.h File Reference	2849
7.248 inst/include/Rcpp/sugar/functions/ifelse.h File Reference	2850
7.249 inst/include/Rcpp/sugar/functions/is_finite.h File Reference	2851
7.250 inst/include/Rcpp/traits/is_finite.h File Reference	2851
7.251 inst/include/Rcpp/sugar/functions/is_infinite.h File Reference	2852
7.252 inst/include/Rcpp/traits/is_infinite.h File Reference	2853
7.253 inst/include/Rcpp/sugar/functions/is_na.h File Reference	2853
7.254 inst/include/Rcpp/traits/is_na.h File Reference	2854
7.255 inst/include/Rcpp/sugar/functions/is_nan.h File Reference	2854
7.256 inst/include/Rcpp/traits/is_nan.h File Reference	2855
7.257 inst/include/Rcpp/sugar/functions/lapply.h File Reference	2856

---

7.258 inst/include/Rcpp/sugar/functions/Lazy.h File Reference . . . . .	2856
7.259 inst/include/Rcpp/sugar/functions/mapply.h File Reference . . . . .	2857
7.260 inst/include/Rcpp/sugar/functions/mapply/mapply_2.h File Reference . . . . .	2857
7.261 inst/include/Rcpp/sugar/functions/mapply/mapply_3.h File Reference . . . . .	2858
7.262 inst/include/Rcpp/sugar/functions/match.h File Reference . . . . .	2859
7.263 inst/include/Rcpp/sugar/functions/math.h File Reference . . . . .	2859
7.263.1 Function Documentation . . . . .	2860
7.263.1.1 VECTORIZED_MATH_1() . . . . .	2860
7.264 inst/include/Rcpp/sugar/functions/max.h File Reference . . . . .	2860
7.265 inst/include/Rcpp/sugar/functions/mean.h File Reference . . . . .	2860
7.266 inst/include/Rcpp/sugar/functions/median.h File Reference . . . . .	2861
7.267 inst/include/Rcpp/sugar/functions/min.h File Reference . . . . .	2862
7.268 inst/include/Rcpp/sugar/functions/na_omit.h File Reference . . . . .	2863
7.269 inst/include/Rcpp/sugar/functions/pmax.h File Reference . . . . .	2863
7.270 inst/include/Rcpp/sugar/functions/pmin.h File Reference . . . . .	2864
7.271 inst/include/Rcpp/sugar/functions/pow.h File Reference . . . . .	2865
7.272 inst/include/Rcpp/sugar/functions/range.h File Reference . . . . .	2865
7.273 inst/include/Rcpp/sugar/functions/rep.h File Reference . . . . .	2866
7.274 inst/include/Rcpp/sugar/functions/rep_each.h File Reference . . . . .	2867
7.275 inst/include/Rcpp/sugar/functions/rep_len.h File Reference . . . . .	2867
7.276 inst/include/Rcpp/sugar/functions/rev.h File Reference . . . . .	2868
7.277 inst/include/Rcpp/sugar/functions/rowSums.h File Reference . . . . .	2868
7.277.1 Macro Definition Documentation . . . . .	2870
7.277.1.1 COL_MEANS_IMPL_KEEPNA . . . . .	2870
7.277.1.2 COL_MEANS_IMPL_RMNA . . . . .	2870
7.277.1.3 COL_SUMS_IMPL_KEEPNA . . . . .	2871
7.277.1.4 COL_SUMS_IMPL_RMNA . . . . .	2871
7.277.1.5 ROW_MEANS_IMPL_KEEPNA . . . . .	2871
7.277.1.6 ROW_MEANS_IMPL_RMNA . . . . .	2871
7.277.1.7 ROW_SUMS_IMPL_KEEPNA . . . . .	2871
7.277.1.8 ROW_SUMS_IMPL_RMNA . . . . .	2872
7.278 inst/include/Rcpp/sugar/functions/sample.h File Reference . . . . .	2872
7.279 inst/include/Rcpp/sugar/functions/sapply.h File Reference . . . . .	2873
7.280 inst/include/Rcpp/sugar/functions/sd.h File Reference . . . . .	2874
7.281 inst/include/Rcpp/sugar/functions/self_match.h File Reference . . . . .	2874
7.282 inst/include/Rcpp/sugar/functions/seq_along.h File Reference . . . . .	2875
7.283 inst/include/Rcpp/sugar/functions/setdiff.h File Reference . . . . .	2876
7.284 inst/include/Rcpp/sugar/functions/sign.h File Reference . . . . .	2876
7.285 inst/include/Rcpp/sugar/functions/strings/collapse.h File Reference . . . . .	2877

7.286 inst/include/Rcpp/sugar/functions/strings/strings.h File Reference . . . . .	2878
7.287 inst/include/Rcpp/sugar/functions/strings/trimws.h File Reference . . . . .	2878
7.288 inst/include/Rcpp/sugar/functions/sum.h File Reference . . . . .	2879
7.289 inst/include/Rcpp/sugar/functions/table.h File Reference . . . . .	2880
7.290 inst/include/Rcpp/sugar/functions/tail.h File Reference . . . . .	2881
7.291 inst/include/Rcpp/sugar/functions/unique.h File Reference . . . . .	2881
7.292 inst/include/Rcpp/sugar/functions/var.h File Reference . . . . .	2882
7.293 inst/include/Rcpp/sugar/functions/which_max.h File Reference . . . . .	2883
7.294 inst/include/Rcpp/sugar/functions/which_min.h File Reference . . . . .	2883
7.295 inst/include/Rcpp/sugar/logical/and.h File Reference . . . . .	2884
7.295.1 Function Documentation . . . . .	2885
7.295.1.1 operator&() . . . . .	2885
7.295.1.2 operator&&() [1/3] . . . . .	2885
7.295.1.3 operator&&() [2/3] . . . . .	2885
7.295.1.4 operator&&() [3/3] . . . . .	2885
7.296 inst/include/Rcpp/sugar/logical/can_have_na.h File Reference . . . . .	2886
7.297 inst/include/Rcpp/sugar/logical/logical.h File Reference . . . . .	2886
7.298 inst/include/Rcpp/sugar/logical/not.h File Reference . . . . .	2887
7.298.1 Function Documentation . . . . .	2887
7.298.1.1 operator"!()" . . . . .	2887
7.299 inst/include/Rcpp/sugar/operators/not.h File Reference . . . . .	2888
7.299.1 Function Documentation . . . . .	2888
7.299.1.1 operator"!()" . . . . .	2888
7.300 inst/include/Rcpp/sugar/logical/or.h File Reference . . . . .	2889
7.300.1 Function Documentation . . . . .	2890
7.300.1.1 operator"   () . . . . .	2890
7.300.1.2 operator"   "   () [1/3] . . . . .	2890
7.300.1.3 operator"   "   () [2/3] . . . . .	2890
7.300.1.4 operator"   "   () [3/3] . . . . .	2890
7.301 inst/include/Rcpp/sugar/logical/SingleLogicalResult.h File Reference . . . . .	2891
7.302 inst/include/Rcpp/sugar/matrix/as_vector.h File Reference . . . . .	2891
7.303 inst/include/Rcpp/sugar/matrix/col.h File Reference . . . . .	2892
7.304 inst/include/Rcpp/sugar/matrix/diag.h File Reference . . . . .	2892
7.305 inst/include/Rcpp/sugar/matrix/lower_tri.h File Reference . . . . .	2893
7.306 inst/include/Rcpp/sugar/matrix/matrix_functions.h File Reference . . . . .	2893
7.307 inst/include/Rcpp/sugar/matrix/outer.h File Reference . . . . .	2894
7.308 inst/include/Rcpp/sugar/matrix/row.h File Reference . . . . .	2894
7.309 inst/include/Rcpp/sugar/matrix/tools.h File Reference . . . . .	2895
7.310 inst/include/Rcpp/sugar/matrix/upper_tri.h File Reference . . . . .	2896

7.311	inst/include/Rcpp/sugar/nona/nona.h File Reference	2896
7.312	inst/include/Rcpp/sugar/operators/Comparator.h File Reference	2897
7.313	inst/include/Rcpp/sugar/operators/Comparator_With_One_Value.h File Reference	2897
7.314	inst/include/Rcpp/sugar/operators/divides.h File Reference	2898
7.315	inst/include/Rcpp/sugar/operators/logical_operators__Vector__primitive.h File Reference	2899
7.315.1	Function Documentation	2900
7.315.1.1	operator"!="() [1/2]	2900
7.315.1.2	operator"!="() [2/2]	2900
7.315.1.3	operator"<()" [1/2]	2901
7.315.1.4	operator"<()" [2/2]	2901
7.315.1.5	operator"<="() [1/2]	2901
7.315.1.6	operator"<="() [2/2]	2901
7.315.1.7	operator"=="() [1/2]	2902
7.315.1.8	operator"=="() [2/2]	2902
7.315.1.9	operator">()" [1/2]	2902
7.315.1.10	operator">()" [2/2]	2902
7.315.1.11	operator">="() [1/2]	2903
7.315.1.12	operator">="() [2/2]	2903
7.316	inst/include/Rcpp/sugar/operators/logical_operators__Vector__Vector.h File Reference	2903
7.316.1	Function Documentation	2904
7.316.1.1	operator"!="()	2904
7.316.1.2	operator"<()"	2904
7.316.1.3	operator"<="()	2905
7.316.1.4	operator"=="()	2905
7.316.1.5	operator">()"	2905
7.316.1.6	operator">="()	2905
7.317	inst/include/Rcpp/sugar/operators/minus.h File Reference	2906
7.318	inst/include/Rcpp/sugar/operators/operators.h File Reference	2907
7.319	inst/include/Rcpp/sugar/operators/plus.h File Reference	2907
7.320	inst/include/Rcpp/sugar/operators/r_binary_op.h File Reference	2908
7.320.1	Macro Definition Documentation	2909
7.320.1.1	Rcpp_OP	2909
7.321	inst/include/Rcpp/sugar/operators/times.h File Reference	2909
7.322	inst/include/Rcpp/sugar/operators/unary_minus.h File Reference	2911
7.322.1	Function Documentation	2911
7.322.1.1	operator"-"()	2911
7.323	inst/include/Rcpp/sugar/Range.h File Reference	2912
7.324	inst/include/Rcpp/sugar/sets.h File Reference	2912
7.324.1	Macro Definition Documentation	2912

---

7.324.1.1 RCPP_UNORDERED_MAP	2912
7.324.1.2 RCPP_UNORDERED_SET	2912
7.325 inst/include/Rcpp/sugar/sugar.h File Reference	2913
7.326 inst/include/Rcpp/sugar/sugar_forward.h File Reference	2913
7.327 inst/include/Rcpp/sugar/tools/iterator.h File Reference	2914
7.328 inst/include/Rcpp/sugar/undoRmath.h File Reference	2914
7.329 inst/include/Rcpp/Symbol.h File Reference	2915
7.330 inst/include/Rcpp/traits/char_type.h File Reference	2915
7.331 inst/include/Rcpp/traits/enable_if.h File Reference	2916
7.332 inst/include/Rcpp/traits/expands_to_logical.h File Reference	2916
7.333 inst/include/Rcpp/traits/get_na.h File Reference	2917
7.334 inst/include/Rcpp/traits/has_iterator.h File Reference	2917
7.335 inst/include/Rcpp/traits/has_na.h File Reference	2918
7.336 inst/include/Rcpp/traits/if_.h File Reference	2919
7.337 inst/include/Rcpp/traits/init_type.h File Reference	2919
7.338 inst/include/Rcpp/traits/integral_constant.h File Reference	2920
7.339 inst/include/Rcpp/traits/is_arithmetic.h File Reference	2920
7.340 inst/include/Rcpp/traits/is_bool.h File Reference	2921
7.341 inst/include/Rcpp/traits/is_const.h File Reference	2922
7.342 inst/include/Rcpp/traits/is_convertible.h File Reference	2922
7.343 inst/include/Rcpp/traits/is_eigen_base.h File Reference	2923
7.344 inst/include/Rcpp/traits/is_module_object.h File Reference	2923
7.345 inst/include/Rcpp/traits/is_pointer.h File Reference	2924
7.346 inst/include/Rcpp/traits/is_primitive.h File Reference	2924
7.347 inst/include/Rcpp/traits/is_reference.h File Reference	2925
7.348 inst/include/Rcpp/traits/is_sugar_expression.h File Reference	2925
7.349 inst/include/Rcpp/traits/is_trivial.h File Reference	2926
7.350 inst/include/Rcpp/traits/is_wide_string.h File Reference	2926
7.351 inst/include/Rcpp/traits/matrix_interface.h File Reference	2927
7.352 inst/include/Rcpp/traits/module_wrap_traits.h File Reference	2927
7.353 inst/include/Rcpp/traits/named_object.h File Reference	2928
7.354 inst/include/Rcpp/traits/num2type.h File Reference	2929
7.355 inst/include/Rcpp/traits/one_type.h File Reference	2929
7.356 inst/include/Rcpp/traits/r_sexptype_traits.h File Reference	2929
7.357 inst/include/Rcpp/traits/r_type_traits.h File Reference	2930
7.358 inst/include/Rcpp/traits/remove_const.h File Reference	2932
7.359 inst/include/Rcpp/traits/remove_const_and_reference.h File Reference	2933
7.360 inst/include/Rcpp/traits/remove_reference.h File Reference	2933
7.361 inst/include/Rcpp/traits/result_of.h File Reference	2934

---

7.362 inst/include/Rcpp/traits/same_type.h File Reference . . . . .	2934
7.363 inst/include/Rcpp/traits/storage_type.h File Reference . . . . .	2935
7.364 inst/include/Rcpp/traits/un_pointer.h File Reference . . . . .	2935
7.365 inst/include/Rcpp/traits/wrap_type_traits.h File Reference . . . . .	2936
7.366 inst/include/Rcpp/unwindProtect.h File Reference . . . . .	2937
7.367 inst/include/Rcpp/utils/tinyformat/tinyformat.h File Reference . . . . .	2938
7.367.1 Macro Definition Documentation . . . . .	2941
7.367.1.1 TINYFORMAT_ARGTYPES . . . . .	2941
7.367.1.2 TINYFORMAT_ARGTYPES_1 . . . . .	2942
7.367.1.3 TINYFORMAT_ARGTYPES_10 . . . . .	2942
7.367.1.4 TINYFORMAT_ARGTYPES_11 . . . . .	2942
7.367.1.5 TINYFORMAT_ARGTYPES_12 . . . . .	2942
7.367.1.6 TINYFORMAT_ARGTYPES_13 . . . . .	2942
7.367.1.7 TINYFORMAT_ARGTYPES_14 . . . . .	2943
7.367.1.8 TINYFORMAT_ARGTYPES_15 . . . . .	2943
7.367.1.9 TINYFORMAT_ARGTYPES_16 . . . . .	2943
7.367.1.10 TINYFORMAT_ARGTYPES_2 . . . . .	2943
7.367.1.11 TINYFORMAT_ARGTYPES_3 . . . . .	2943
7.367.1.12 TINYFORMAT_ARGTYPES_4 . . . . .	2944
7.367.1.13 TINYFORMAT_ARGTYPES_5 . . . . .	2944
7.367.1.14 TINYFORMAT_ARGTYPES_6 . . . . .	2944
7.367.1.15 TINYFORMAT_ARGTYPES_7 . . . . .	2944
7.367.1.16 TINYFORMAT_ARGTYPES_8 . . . . .	2944
7.367.1.17 TINYFORMAT_ARGTYPES_9 . . . . .	2945
7.367.1.18 TINYFORMAT_ASSERT . . . . .	2945
7.367.1.19 TINYFORMAT_DEFINE_FORMAT_TRUNCATED_CSTR . . . . .	2945
7.367.1.20 TINYFORMAT_DEFINE_FORMATVALUE_CHAR . . . . .	2945
7.367.1.21 TINYFORMAT_ERROR . . . . .	2946
7.367.1.22 TINYFORMAT_FOREACH_ARGNUM . . . . .	2946
7.367.1.23 TINYFORMAT_HIDDEN . . . . .	2946
7.367.1.24 TINYFORMAT_MAKE_FORMAT_FUNCS . . . . .	2946
7.367.1.25 TINYFORMAT_MAKE_FORMATLIST_CONSTRUCTOR . . . . .	2947
7.367.1.26 TINYFORMAT_MAKE_MAKEFORMATLIST . . . . .	2947
7.367.1.27 TINYFORMAT_PASSARGS . . . . .	2947
7.367.1.28 TINYFORMAT_PASSARGS_1 . . . . .	2947
7.367.1.29 TINYFORMAT_PASSARGS_10 . . . . .	2948
7.367.1.30 TINYFORMAT_PASSARGS_11 . . . . .	2948
7.367.1.31 TINYFORMAT_PASSARGS_12 . . . . .	2948
7.367.1.32 TINYFORMAT_PASSARGS_13 . . . . .	2948



---

7.367.1.33 TINYFORMAT_PASSARGS_14	2948
7.367.1.34 TINYFORMAT_PASSARGS_15	2948
7.367.1.35 TINYFORMAT_PASSARGS_16	2949
7.367.1.36 TINYFORMAT_PASSARGS_2	2949
7.367.1.37 TINYFORMAT_PASSARGS_3	2949
7.367.1.38 TINYFORMAT_PASSARGS_4	2949
7.367.1.39 TINYFORMAT_PASSARGS_5	2949
7.367.1.40 TINYFORMAT_PASSARGS_6	2949
7.367.1.41 TINYFORMAT_PASSARGS_7	2950
7.367.1.42 TINYFORMAT_PASSARGS_8	2950
7.367.1.43 TINYFORMAT_PASSARGS_9	2950
7.367.1.44 TINYFORMAT_PASSARGS_TAIL	2950
7.367.1.45 TINYFORMAT_PASSARGS_TAIL_1	2950
7.367.1.46 TINYFORMAT_PASSARGS_TAIL_10	2950
7.367.1.47 TINYFORMAT_PASSARGS_TAIL_11	2951
7.367.1.48 TINYFORMAT_PASSARGS_TAIL_12	2951
7.367.1.49 TINYFORMAT_PASSARGS_TAIL_13	2951
7.367.1.50 TINYFORMAT_PASSARGS_TAIL_14	2951
7.367.1.51 TINYFORMAT_PASSARGS_TAIL_15	2951
7.367.1.52 TINYFORMAT_PASSARGS_TAIL_16	2951
7.367.1.53 TINYFORMAT_PASSARGS_TAIL_2	2952
7.367.1.54 TINYFORMAT_PASSARGS_TAIL_3	2952
7.367.1.55 TINYFORMAT_PASSARGS_TAIL_4	2952
7.367.1.56 TINYFORMAT_PASSARGS_TAIL_5	2952
7.367.1.57 TINYFORMAT_PASSARGS_TAIL_6	2952
7.367.1.58 TINYFORMAT_PASSARGS_TAIL_7	2952
7.367.1.59 TINYFORMAT_PASSARGS_TAIL_8	2953
7.367.1.60 TINYFORMAT_PASSARGS_TAIL_9	2953
7.367.1.61 TINYFORMAT_VARARGS	2953
7.367.1.62 TINYFORMAT_VARARGS_1	2953
7.367.1.63 TINYFORMAT_VARARGS_10	2953
7.367.1.64 TINYFORMAT_VARARGS_11	2954
7.367.1.65 TINYFORMAT_VARARGS_12	2954
7.367.1.66 TINYFORMAT_VARARGS_13	2954
7.367.1.67 TINYFORMAT_VARARGS_14	2954
7.367.1.68 TINYFORMAT_VARARGS_15	2954
7.367.1.69 TINYFORMAT_VARARGS_16	2955
7.367.1.70 TINYFORMAT_VARARGS_2	2955
7.367.1.71 TINYFORMAT_VARARGS_3	2955

---

7.367.1.72 TINYFORMAT_VARARGS_4	2955
7.367.1.73 TINYFORMAT_VARARGS_5	2955
7.367.1.74 TINYFORMAT_VARARGS_6	2956
7.367.1.75 TINYFORMAT_VARARGS_7	2956
7.367.1.76 TINYFORMAT_VARARGS_8	2956
7.367.1.77 TINYFORMAT_VARARGS_9	2956
7.368 inst/include/Rcpp/utis/tinyformat.h File Reference	2957
7.368.1 Macro Definition Documentation	2958
7.368.1.1 TINYFORMAT_ASSERT	2958
7.368.1.2 TINYFORMAT_ERROR	2958
7.368.1.3 TINYFORMAT_NO_VARIADIC_TEMPLATES	2958
7.369 inst/include/Rcpp/vector/00_forward_proxy.h File Reference	2958
7.370 inst/include/Rcpp/vector/00_forward_Vector.h File Reference	2959
7.371 inst/include/Rcpp/vector/ChildVector.h File Reference	2959
7.372 inst/include/Rcpp/vector/const_generic_proxy.h File Reference	2959
7.373 inst/include/Rcpp/vector/const_string_proxy.h File Reference	2960
7.374 inst/include/Rcpp/vector/converter.h File Reference	2961
7.375 inst/include/Rcpp/vector/DimNameProxy.h File Reference	2961
7.376 inst/include/Rcpp/vector/generic_proxy.h File Reference	2962
7.377 inst/include/Rcpp/vector/instantiation.h File Reference	2962
7.378 inst/include/Rcpp/vector/LazyVector.h File Reference	2963
7.379 inst/include/Rcpp/vector/ListOf.h File Reference	2964
7.380 inst/include/Rcpp/vector/Matrix.h File Reference	2964
7.380.1 Macro Definition Documentation	2965
7.380.1.1 RCPP_GENERATE_MATRIX_SCALAR_OPERATOR	2965
7.380.1.2 RCPP_GENERATE_SCALAR_MATRIX_OPERATOR	2966
7.381 inst/include/Rcpp/vector/MatrixBase.h File Reference	2966
7.382 inst/include/Rcpp/vector/MatrixColumn.h File Reference	2967
7.383 inst/include/Rcpp/vector/MatrixRow.h File Reference	2967
7.384 inst/include/Rcpp/vector/no_init.h File Reference	2968
7.385 inst/include/Rcpp/vector/RangeIndexer.h File Reference	2968
7.385.1 Macro Definition Documentation	2969
7.385.1.1 UNROLL_LOOP	2969
7.386 inst/include/Rcpp/vector/string_proxy.h File Reference	2970
7.387 inst/include/Rcpp/vector/SubMatrix.h File Reference	2970
7.387.1 Macro Definition Documentation	2971
7.387.1.1 RCPP_WRAP_SUBMATRIX	2971
7.388 inst/include/Rcpp/vector/Subsetter.h File Reference	2971
7.388.1 Macro Definition Documentation	2972

---

7.388.1.1 RCPP_GENERATE_SUBSET_PROXY_OPERATOR . . . . .	2972
7.389 inst/include/Rcpp/vector/swap.h File Reference . . . . .	2972
7.389.1 Macro Definition Documentation . . . . .	2973
7.389.1.1 RCPP_GENERATE_SWAP . . . . .	2973
7.390 inst/include/Rcpp/vector/vector_from_string.h File Reference . . . . .	2973
7.391 inst/include/Rcpp/vector/VectorBase.h File Reference . . . . .	2974
7.392 inst/include/Rcpp/WeakReference.h File Reference . . . . .	2974
7.393 inst/include/Rcpp/XPtr.h File Reference . . . . .	2975
7.394 inst/include/RcppCommon.h File Reference . . . . .	2976
7.394.1 Macro Definition Documentation . . . . .	2977
7.394.1.1 RcppExport . . . . .	2978
7.395 src/api.cpp File Reference . . . . .	2978
7.395.1 Macro Definition Documentation . . . . .	2979
7.395.1.1 COMPILING_RCPP . . . . .	2979
7.395.1.2 RCPP_USE_GLOBAL_ROSTREAM . . . . .	2979
7.395.2 Function Documentation . . . . .	2979
7.395.2.1 as_character_externalptr() . . . . .	2979
7.395.2.2 demangle() . . . . .	2980
7.395.2.3 getRcppVersionStrings() . . . . .	2980
7.395.2.4 rcpp_can_use_cxx0x() . . . . .	2980
7.395.2.5 rcpp_can_use_cxx11() . . . . .	2980
7.395.2.6 rcpp_capabilities() . . . . .	2980
7.395.2.7 short_file_name() . . . . .	2981
7.395.2.8 stack_trace() . . . . .	2981
7.396 src/attributes.cpp File Reference . . . . .	2981
7.396.1 Macro Definition Documentation . . . . .	2983
7.396.1.1 COMPILING_RCPP . . . . .	2983
7.396.1.2 RCPP_NO_SUGAR . . . . .	2983
7.396.2 Function Documentation . . . . .	2983
7.396.2.1 compileAttributes() . . . . .	2984
7.396.2.2 sourceCppContext() . . . . .	2985
7.397 src/barrier.cpp File Reference . . . . .	2985
7.397.1 Macro Definition Documentation . . . . .	2987
7.397.1.1 COMPILING_RCPP . . . . .	2987
7.397.1.2 RCPP_CACHE_SIZE . . . . .	2987
7.397.1.3 RCPP_HASH_CACHE_INDEX . . . . .	2987
7.397.1.4 RCPP_HASH_CACHE_INITIAL_SIZE . . . . .	2987
7.397.1.5 USE_RINTERNALS . . . . .	2987
7.397.2 Function Documentation . . . . .	2988

---

7.397.2.1	<a href="#">char_get_string_elt()</a>	2988
7.397.2.2	<a href="#">char_nocheck()</a>	2988
7.397.2.3	<a href="#">char_set_string_elt()</a>	2988
7.397.2.4	<a href="#">dataptr()</a>	2989
7.397.2.5	<a href="#">error_occured()</a>	2989
7.397.2.6	<a href="#">get_cache()</a>	2989
7.397.2.7	<a href="#">get_rcpp_cache()</a>	2990
7.397.2.8	<a href="#">get_string_elt()</a>	2990
7.397.2.9	<a href="#">get_string_ptr()</a>	2990
7.397.2.10	<a href="#">get_vector_elt()</a>	2990
7.397.2.11	<a href="#">get_vector_ptr()</a>	2991
7.397.2.12	<a href="#">init_Rcpp_cache()</a>	2991
7.397.2.13	<a href="#">rcpp_error_recorder()</a>	2992
7.397.2.14	<a href="#">rcpp_get_current_error()</a>	2992
7.397.2.15	<a href="#">rcpp_get_stack_trace()</a>	2993
7.397.2.16	<a href="#">rcpp_set_stack_trace()</a>	2993
7.397.2.17	<a href="#">reset_current_error()</a>	2994
7.397.2.18	<a href="#">set_current_error()</a>	2994
7.397.2.19	<a href="#">set_error_occured()</a>	2994
7.397.2.20	<a href="#">set_string_elt()</a>	2995
7.397.2.21	<a href="#">set_vector_elt()</a>	2995
7.397.3	<a href="#">Variable Documentation</a>	2995
7.397.3.1	<a href="#">Rcpp_cache</a>	2995
7.397.3.2	<a href="#">Rcpp_cache_know</a>	2995
7.398	<a href="#">src/date.cpp File Reference</a>	2996
7.398.1	<a href="#">Macro Definition Documentation</a>	2998
7.398.1.1	<a href="#">_NO_OLDNAMES</a>	2999
7.398.1.2	<a href="#">AVGSECSPERYEAR</a>	2999
7.398.1.3	<a href="#">BIGGEST</a>	2999
7.398.1.4	<a href="#">COMPILING_RCPP</a>	2999
7.398.1.5	<a href="#">DAY_OF_YEAR</a>	2999
7.398.1.6	<a href="#">days_in_year</a>	3000
7.398.1.7	<a href="#">DAYSPLYEAR</a>	3000
7.398.1.8	<a href="#">DAYSPLYEAR</a>	3000
7.398.1.9	<a href="#">DAYSPLYEAR</a>	3000
7.398.1.10	<a href="#">E_OVERFLOW</a>	3000
7.398.1.11	<a href="#">EPOCH_WDAY</a>	3000
7.398.1.12	<a href="#">EPOCH_YEAR</a>	3001
7.398.1.13	<a href="#">gmtptr</a>	3001

---

7.398.1.14 GRANDPARENTED	3001
7.398.1.15 HOURS PER DAY	3001
7.398.1.16 INITIALIZE	3001
7.398.1.17 is_digit	3002
7.398.1.18 isleap [1/2]	3002
7.398.1.19 isleap [2/2]	3002
7.398.1.20 isleap_sum	3002
7.398.1.21 JULIAN_DAY	3002
7.398.1.22 MAXVAL	3003
7.398.1.23 MINS PER HOUR	3003
7.398.1.24 MINVAL	3003
7.398.1.25 MONS PER YEAR	3003
7.398.1.26 MONTH_NTH_DAY_OF_WEEK	3003
7.398.1.27 MY_TZNAME_MAX	3004
7.398.1.28 OPEN_MODE	3004
7.398.1.29 SECS PER DAY	3004
7.398.1.30 SECS PER HOUR	3004
7.398.1.31 SECS PER MIN	3004
7.398.1.32 SECS PER REPEAT	3004
7.398.1.33 SECS PER REPEAT_BITS	3005
7.398.1.34 TM_APRIL	3005
7.398.1.35 TM_AUGUST	3005
7.398.1.36 TM_DECEMBER	3005
7.398.1.37 TM_FEBRUARY	3005
7.398.1.38 TM_FRIDAY	3005
7.398.1.39 TM_JANUARY	3006
7.398.1.40 TM_JULY	3006
7.398.1.41 TM_JUNE	3006
7.398.1.42 TM_MARCH	3006
7.398.1.43 TM_MAY	3006
7.398.1.44 TM_MONDAY	3006
7.398.1.45 TM_NOVEMBER	3007
7.398.1.46 TM_OCTOBER	3007
7.398.1.47 TM_SATURDAY	3007
7.398.1.48 TM_SEPTEMBER	3007
7.398.1.49 TM_SUNDAY	3007
7.398.1.50 TM_THURSDAY	3007
7.398.1.51 TM_TUESDAY	3008
7.398.1.52 TM_WEDNESDAY	3008

---

7.398.1.53	TM_YEAR_BASE	3008
7.398.1.54	TWOS_COMPLEMENT	3008
7.398.1.55	TYPE_BIT	3008
7.398.1.56	TYPE_INTEGRAL	3009
7.398.1.57	TYPE_SIGNED	3009
7.398.1.58	TZ_MAGIC	3009
7.398.1.59	TZ_MAX_CHARS	3009
7.398.1.60	TZ_MAX_LEAPS	3009
7.398.1.61	TZ_MAX_TIMES	3010
7.398.1.62	TZ_MAX_TYPES	3010
7.398.1.63	TZDEFAULT	3010
7.398.1.64	TZDEFRULES	3010
7.398.1.65	TZDEFRULESTRING	3010
7.398.1.66	TZDIR	3010
7.398.1.67	TZFILE_H	3011
7.398.1.68	YEARSPPERPEAT	3011
7.399	src/internal.h File Reference	3011
7.399.1	Macro Definition Documentation	3013
7.399.1.1	CALLFUN_0	3013
7.399.1.2	CALLFUN_1	3013
7.399.1.3	CALLFUN_2	3013
7.399.1.4	CALLFUN_3	3013
7.399.1.5	CALLFUN_4	3013
7.399.1.6	CALLFUN_5	3014
7.399.1.7	EXTFUN	3014
7.399.1.8	MAX_ARGS	3014
7.399.1.9	RCPP_FUN_1	3014
7.399.1.10	RCPP_FUN_2	3015
7.399.1.11	RCPP_FUN_3	3015
7.399.1.12	RCPP_FUN_4	3016
7.399.1.13	UNPACK_EXTERNAL_ARGS	3016
7.399.2	Function Documentation	3016
7.399.2.1	CALLFUN_0() [1/6]	3016
7.399.2.2	CALLFUN_0() [2/6]	3017
7.399.2.3	CALLFUN_0() [3/6]	3017
7.399.2.4	CALLFUN_0() [4/6]	3017
7.399.2.5	CALLFUN_0() [5/6]	3017
7.399.2.6	CALLFUN_0() [6/6]	3017
7.399.2.7	CALLFUN_1() [1/11]	3017

---

7.399.2.8 CALLFUN_1() [2/11]	3018
7.399.2.9 CALLFUN_1() [3/11]	3018
7.399.2.10 CALLFUN_1() [4/11]	3018
7.399.2.11 CALLFUN_1() [5/11]	3018
7.399.2.12 CALLFUN_1() [6/11]	3018
7.399.2.13 CALLFUN_1() [7/11]	3018
7.399.2.14 CALLFUN_1() [8/11]	3019
7.399.2.15 CALLFUN_1() [9/11]	3019
7.399.2.16 CALLFUN_1() [10/11]	3019
7.399.2.17 CALLFUN_1() [11/11]	3019
7.399.2.18 CALLFUN_2() [1/5]	3019
7.399.2.19 CALLFUN_2() [2/5]	3019
7.399.2.20 CALLFUN_2() [3/5]	3020
7.399.2.21 CALLFUN_2() [4/5]	3020
7.399.2.22 CALLFUN_2() [5/5]	3020
7.399.2.23 CALLFUN_3()	3020
7.399.2.24 CALLFUN_4()	3020
7.399.2.25 EXTFUN() [1/7]	3020
7.399.2.26 EXTFUN() [2/7]	3021
7.399.2.27 EXTFUN() [3/7]	3021
7.399.2.28 EXTFUN() [4/7]	3021
7.399.2.29 EXTFUN() [5/7]	3021
7.399.2.30 EXTFUN() [6/7]	3021
7.399.2.31 EXTFUN() [7/7]	3021
7.399.2.32 get_Rcpp_protection_stack()	3022
7.399.2.33 init_Rcpp_routines()	3022
7.400 src/module.cpp File Reference	3022
7.400.1 Macro Definition Documentation	3023
7.400.1.1 CHECK_DUMMY_OBJ	3024
7.400.1.2 COMPILING_RCPP	3024
7.400.2 Typedef Documentation	3024
7.400.2.1 XP_Class	3024
7.400.2.2 XP_Function	3024
7.400.2.3 XP_Module	3024
7.400.3 Function Documentation	3024
7.400.3.1 class__dummyInstance()	3025
7.400.3.2 class__newInstance()	3025
7.400.3.3 CppMethod__invoke()	3025
7.400.3.4 CppMethod__invoke_notvoid()	3025

---

7.400.3.5 CppMethod__invoke_void()	3026
7.400.3.6 getCurrentScope()	3026
7.400.3.7 InternalFunction_invoke()	3026
7.400.3.8 Module__invoke()	3026
7.400.3.9 RCPP_FUN_1() [1/14]	3027
7.400.3.10 RCPP_FUN_1() [2/14]	3027
7.400.3.11 RCPP_FUN_1() [3/14]	3027
7.400.3.12 RCPP_FUN_1() [4/14]	3027
7.400.3.13 RCPP_FUN_1() [5/14]	3028
7.400.3.14 RCPP_FUN_1() [6/14]	3028
7.400.3.15 RCPP_FUN_1() [7/14]	3028
7.400.3.16 RCPP_FUN_1() [8/14]	3028
7.400.3.17 RCPP_FUN_1() [9/14]	3029
7.400.3.18 RCPP_FUN_1() [10/14]	3029
7.400.3.19 RCPP_FUN_1() [11/14]	3029
7.400.3.20 RCPP_FUN_1() [12/14]	3029
7.400.3.21 RCPP_FUN_1() [13/14]	3030
7.400.3.22 RCPP_FUN_1() [14/14]	3030
7.400.3.23 RCPP_FUN_2() [1/9]	3030
7.400.3.24 RCPP_FUN_2() [2/9]	3030
7.400.3.25 RCPP_FUN_2() [3/9]	3031
7.400.3.26 RCPP_FUN_2() [4/9]	3031
7.400.3.27 RCPP_FUN_2() [5/9]	3031
7.400.3.28 RCPP_FUN_2() [6/9]	3031
7.400.3.29 RCPP_FUN_2() [7/9]	3032
7.400.3.30 RCPP_FUN_2() [8/9]	3032
7.400.3.31 RCPP_FUN_2() [9/9]	3032
7.400.3.32 RCPP_FUN_3()	3032
7.400.3.33 RCPP_FUN_4()	3033
7.400.3.34 setCurrentScope()	3033
7.400.4 Variable Documentation	3033
7.400.4.1 rcpp_dummy_pointer	3033
7.401 src/rcpp_init.cpp File Reference	3033
7.401.1 Macro Definition Documentation	3034
7.401.1.1 CALLDEF	3034
7.401.1.2 COMPILING_RCPP	3034
7.401.1.3 EXTDEF	3035
7.401.1.4 RCPP_REGISTER	3035
7.401.2 Function Documentation	3035



7.401.2.1	<a href="#">init_Rcpp_routines()</a>	3035
7.401.2.2	<a href="#">R_init_Rcpp()</a>	3035
7.401.2.3	<a href="#">R_unload_Rcpp()</a>	3037
7.401.2.4	<a href="#">registerFunctions()</a>	3037
7.401.3	<a href="#">Variable Documentation</a>	3038
7.401.3.1	<a href="#">callEntries</a>	3039
7.401.3.2	<a href="#">extEntries</a>	3039
<b>8</b>	<b><a href="#">Example Documentation</a></b>	<b>3041</b>
8.1	<a href="#">functionCallback/RcppFunctionCallExample.cpp</a>	3041
8.2	<a href="#">functionCallback/ExampleRCode.R</a>	3041
8.3	<a href="#">RcppInline/external_pointer.r</a>	3041
8.4	<a href="#">RcppInline/RcppInlineExample.r</a>	3043
8.5	<a href="#">RcppInline/RcppInlineWithLibsExamples.r</a>	3043
8.6	<a href="#">RcppInline/RObject.r</a>	3047
8.7	<a href="#">RcppInline/RcppSimpleExample.r</a>	3050
8.8	<a href="#">RcppInline/RcppSimpleTests.r</a>	3050
8.9	<a href="#">RcppInline/UncaughtExceptions.r</a>	3051
8.10	<a href="#">ConvolveBenchmarks/convolve2_c.c</a>	3051
8.11	<a href="#">ConvolveBenchmarks/convolve2_cpp.cpp</a>	3052
8.12	<a href="#">ConvolveBenchmarks/convolve3_cpp.cpp</a>	3052
8.13	<a href="#">ConvolveBenchmarks/convolve4_cpp.cpp</a>	3052
8.14	<a href="#">ConvolveBenchmarks/convolve7_c.c</a>	3052
8.15	<a href="#">ConvolveBenchmarks/exampleRCode.r</a>	3053
8.16	<a href="#">FastLM/benchmark.r</a>	3054
8.17	<a href="#">FastLM/fastLMviaArmadillo.r</a>	3055
8.18	<a href="#">FastLM/fastLMviaGSL.r</a>	3055
8.19	<a href="#">FastLM/ImArmadillo.r</a>	3056
8.20	<a href="#">FastLM/ImGSL.r</a>	3056
8.21	<a href="#">SugarPerformance/sugarBenchmarks.R</a>	3056
8.22	<a href="#">SugarPerformance/Timer.h</a>	3059
8.23	<a href="#">SugarPerformance/Timertest.cpp</a>	3060
8.24	<a href="#">OpenMP/piWithInterrupts.cpp.R</a>	3060
8.25	<a href="#">OpenMP/check.R</a>	3060
8.26	<a href="#">OpenMP/OpenMPandInline.r</a>	3060
8.27	<a href="#">RcppGibbs/RcppGibbs.R</a>	3061
8.28	<a href="#">RcppGibbs/timeRNGs.R</a>	3064
8.29	<a href="#">Misc/fibonacci.r</a>	3066
8.30	<a href="#">Misc/ifelseLooped.r</a>	3067



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

R	55
Rcpp	
Rcpp API	94
Rcpp::algorithm	281
Rcpp::algorithm::helpers	287
Rcpp::attributes	288
Rcpp::internal	
Internal implementation details	302
Rcpp::internal::debug	358
Rcpp::InternalFunctionWithStdFunction	359
Rcpp::stats	359
Rcpp::sugar	377
Rcpp::sugar::cbind_impl	439
Rcpp::sugar::cbind_impl::detail	442
Rcpp::sugar::detail	443
Rcpp::sugar::median_detail	448
Rcpp::traits	
Traits used to dispatch wrap	449
std	462
tinyformat	462
tinyformat::detail	467



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Rcpp::traits::_sfnaf_types . . . . .	471
Rcpp::traits::_has_iterator_helper< T > . . . . .	474
Rcpp::traits::_has_matrix_interface_helper< T > . . . . .	476
Rcpp::traits::_has_rtype_helper< T > . . . . .	478
Rcpp::traits::_is_eigen_helper< T > . . . . .	480
Rcpp::traits::_is_exporter_helper< T > . . . . .	482
Rcpp::traits::_is_generator_helper< T > . . . . .	484
Rcpp::traits::_is_importer_helper< T > . . . . .	486
Rcpp::traits::_is_sugar_expression_helper< T > . . . . .	488
Rcpp::traits::_sfnaf_types::_two . . . . .	473
Rcpp::traits::_has_iterator_helper< T >::_Wrap_type< U > . . . . .	491
Rcpp::traits::_has_matrix_interface_helper< T >::_Wrap_type< U > . . . . .	491
Rcpp::traits::_has_rtype_helper< T >::_Wrap_type< U > . . . . .	491
Rcpp::traits::_is_eigen_helper< T >::_Wrap_type< U > . . . . .	492
Rcpp::traits::_is_exporter_helper< T >::_Wrap_type< U > . . . . .	492
Rcpp::traits::_is_generator_helper< T >::_Wrap_type< U > . . . . .	492
Rcpp::traits::_is_importer_helper< T >::_Wrap_type< U > . . . . .	493
Rcpp::traits::_is_sugar_expression_helper< T >::_Wrap_type< U > . . . . .	493
Rcpp::traits::allowed_matrix_type< bool > . . . . .	500
Rcpp::traits::allowed_matrix_type< true > . . . . .	500
Rcpp::Argument . . . . .	538
Rcpp::attributes::Argument . . . . .	539
Rcpp::Armor< T > . . . . .	543
Rcpp::attributes::Attribute . . . . .	548
Rcpp::AttributeProxyPolicy< CLASS > . . . . .	562
Rcpp::Vector< VECSXP, StoragePolicy > . . . . .	2494
Rcpp::DataFrame_Impl< StoragePolicy > . . . . .	936
Rcpp::AttributeProxyPolicy< T > . . . . .	562
Rcpp::ListOf< T > . . . . .	1408
Rcpp::AttributeProxyPolicy< Vector< RTYPE, PreserveStorage > > . . . . .	562

Rcpp::Vector< STRSXP > . . . . .	2494
Rcpp::Vector< VECSXP > . . . . .	2494
Rcpp::Vector< INTSXP > . . . . .	2494
Rcpp::Vector< RTYPE, PreserveStorage > . . . . .	2494
Rcpp::Matrix< RTYPE, StoragePolicy > . . . . .	1462
Rcpp::Vector< RTYPE, StoragePolicy > . . . . .	2494
Rcpp::newDateVector . . . . .	1661
Rcpp::newDatetimeVector . . . . .	1657
Rcpp::AttributeProxyPolicy< XPtr< T, PreserveStorage, standard_delete_finalizer< T >, false > > . . . . .	562
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > . . . . .	2617
Rcpp::traits::is_convertible< T, U >::Big . . . . .	566
std::binary_function	
Rcpp::binary_call< T1, T2, RESULT_TYPE > . . . . .	567
Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E > . . . . .	570
Rcpp::sugar::cbind_impl::BindableExpression< cbind_sexptype_traits< T >::rtype, ScalarBindable< T > > . . . . .	570
Rcpp::sugar::cbind_impl::ScalarBindable< T > . . . . .	2122
Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, ContainerBindable< RTYPE, T > > . . . . .	570
Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T > . . . . .	769
Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, JoinOp< RTYPE, E1, E2 > > . . . . .	570
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 > . . . . .	1379
Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, JoinOp< RTYPE, E1, ScalarBindable< scalar< RTYPE >::type > > > . . . . .	570
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > > > . . . . .	1384
Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, JoinOp< RTYPE, ScalarBindable< scalar< RTYPE >::type >, E2 > > . . . . .	570
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 > > . . . . .	1388
Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, JoinOp< RTYPE, ScalarBindable< scalar< RTYPE >::type >, ScalarBindable< scalar< RTYPE >::type > > > . . . . .	570
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > > > . . . . .	1393
Rcpp::BindingPolicy< EnvironmentClass > . . . . .	579
Cache . . . . .	584
Rcpp::sugar::cbind_impl::cbind_sexptype_traits< SEXP > . . . . .	600
Rcpp::sugar::cbind_impl::cbind_storage_type< LGLSXP > . . . . .	602
Rcpp::traits::char_type< T > . . . . .	602
Rcpp::traits::char_type< const char * > . . . . .	603
Rcpp::traits::char_type< const wchar_t * > . . . . .	604
Rcpp::sugar::clamp_operator< RTYPE, NA > . . . . .	610
Rcpp::sugar::clamp_operator< REALSXP, true > . . . . .	612
class_Base	
class_< Class > . . . . .	618
Rcpp::class_Base . . . . .	638
Rcpp::attributes::CommentState . . . . .	669
Rcpp::MatrixRow< RTYPE >::const_iter_traits . . . . .	714
Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits . . . . .	716
Rcpp::ConstMatrixRow< RTYPE >::const_iterator . . . . .	718
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy > . . . . .	737
Rcpp::ConstInputParameter< T > . . . . .	752
Rcpp::ConstReferenceInputParameter< T > . . . . .	766
Rcpp::traits::container_exporter< Container, double > . . . . .	767
Rcpp::traits::container_exporter< Container, int > . . . . .	768
Rcpp::traits::ContainerExporter< ContainerTemplate, T > . . . . .	774

tinyformat::detail::convertToInt< T, convertible > . . . . .	778
tinyformat::detail::convertToInt< T, true > . . . . .	778
Rcpp::sugar::CountInserter< HASH, STORAGE > . . . . .	779
Rcpp::CppClassFinalizer< Class > . . . . .	797
Rcpp::FunctionFinalizer< Class > . . . . .	1159
Rcpp::CppClassFunctionBase . . . . .	802
Rcpp::CppClassFunction . . . . .	799
Rcpp::InternalFunctionWithStdFunction::CppClassFunctionBaseFromStdFunction< RESULT_TYPE, Args > . . . . .	804
Rcpp::InternalFunctionWithStdFunction::CppClassFunctionBaseFromStdFunction< void, Args... > . . . . .	807
Rcpp::CppClassMethod< Class > . . . . .	819
Rcpp::CppClassInheritedMethod< Class, Parent > . . . . .	809
CppClassProperty	
CppClassProperty_GetConstMethod< Class, PROP > . . . . .	832
CppClassProperty_GetConstMethod_SetMethod< Class, PROP > . . . . .	836
CppClassProperty_GetConstMethod_SetPointer< Class, PROP > . . . . .	841
CppClassProperty_GetMethod< Class, PROP > . . . . .	846
CppClassProperty_GetMethod_SetMethod< Class, PROP > . . . . .	850
CppClassProperty_GetMethod_SetPointer< Class, PROP > . . . . .	855
CppClassProperty_GetPointerMethod< Class, PROP > . . . . .	870
CppClassProperty_GetPointer_SetMethod< Class, PROP > . . . . .	860
CppClassProperty_GetPointer_SetPointer< Class, PROP > . . . . .	865
CppClassProperty_Getter< PROP > . . . . .	874
CppClassProperty_Getter_Setter< PROP > . . . . .	878
Rcpp::CppClassProperty< Class > . . . . .	829
Rcpp::CppClassInheritedProperty< Class, Parent > . . . . .	815
Rcpp::CppClassProperty< Parent > . . . . .	829
Rcpp::algorithm::helpers::ctype< T > . . . . .	882
Rcpp::algorithm::helpers::CTYPE_CHAR . . . . .	885
Rcpp::algorithm::helpers::CTYPE_DOUBLE . . . . .	886
Rcpp::algorithm::helpers::CTYPE_FLOAT . . . . .	887
Rcpp::algorithm::helpers::ctype_helper< I > . . . . .	887
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_CHAR) > . . . . .	888
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_DOUBLE) > . . . . .	889
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_FLOAT) > . . . . .	890
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_INT) > . . . . .	891
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG) > . . . . .	892
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG_DOUBLE) > . . . . .	893
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_SHORT) > . . . . .	894
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_STRING) > . . . . .	895
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) > . . . . .	896
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_INT) > . . . . .	897
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) > . . . . .	898
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) > . . . . .	899
Rcpp::algorithm::helpers::CTYPE_INT . . . . .	900
Rcpp::algorithm::helpers::CTYPE_LONG . . . . .	901
Rcpp::algorithm::helpers::CTYPE_LONG_DOUBLE . . . . .	901
Rcpp::algorithm::helpers::CTYPE_SHORT . . . . .	902
Rcpp::algorithm::helpers::CTYPE_STRING . . . . .	903
Rcpp::algorithm::helpers::CTYPE_UNKNOWN . . . . .	903
Rcpp::algorithm::helpers::CTYPE_UNSIGNED_CHAR . . . . .	904
Rcpp::algorithm::helpers::CTYPE_UNSIGNED_INT . . . . .	905
Rcpp::algorithm::helpers::CTYPE_UNSIGNED_LONG . . . . .	905
Rcpp::algorithm::helpers::CTYPE_UNSIGNED_SHORT . . . . .	906

Rcpp::Date	947
Rcpp::Datetime	956
Rcpp::algorithm::helpers::decays_to_ctype< T >	966
Rcpp::sugar::diag_result_type_trait< T >	977
Rcpp::Dimension	990
Rcpp::internal::DimNameProxy	995
Rcpp::DottedPairImpl< CLASS >	1060
Rcpp::DottedPairProxyPolicy< CLASS >	1072
Rcpp::internal::element_converter< RTYPE >	1073
Rcpp::traits::enable_if< B, T >	1075
Rcpp::traits::enable_if< true, T >	1075
Rcpp::enum_< Enum, Parent >	1076
std::exception	
Rcpp::exception	1080
Rcpp::file_io_error	1129
Rcpp::file_exists	1128
Rcpp::file_not_found	1133
Rcpp::no_such_env	1670
interrupt_exception	1264
Rcpp::algorithm::helpers::exp	1085
Rcpp::traits::expands_to_logical__impl< RTYPE >	1088
Rcpp::MatrixBase< LGLSXP, false, UpperTri< RTYPE, NA, T > >	1494
Rcpp::sugar::UpperTri< RTYPE, NA, T >	2477
Rcpp::MatrixBase< RTYPE, true, SubMatrix< RTYPE > >	1494
Rcpp::SubMatrix< RTYPE >	2293
Rcpp::MatrixBase< INTSXP, false, Row< RTYPE, LHS_NA, LHS_T > >	1494
Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >	2065
Rcpp::MatrixBase< RTYPE, NA, Diag_Maker< RTYPE, NA, T > >	1494
Rcpp::sugar::Diag_Maker< RTYPE, NA, T >	973
Rcpp::MatrixBase< INTSXP, false, Col< RTYPE, LHS_NA, LHS_T > >	1494
Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >	647
Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, ScalarBindable< scalar< RTYPE >::type >, ScalarBindable< scalar< RTYPE >::type > > >	1494
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > > >	1393
Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, ScalarBindable< scalar< RTYPE >::type >, E2 > >	1494
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 > >	1388
Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, E1, ScalarBindable< scalar< RTYPE >::type > > >	1494
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > > >	1384
Rcpp::MatrixBase< LGLSXP, false, LowerTri< RTYPE, NA, T > >	1494
Rcpp::sugar::LowerTri< RTYPE, NA, T >	1433
Rcpp::MatrixBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::traits::result_of< Function >::type >::rtype, true, Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function > >	1494
Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >	1751
Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, E1, E2 > >	1494
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >	1379
Rcpp::MatrixBase< RTYPE, true, Matrix< RTYPE, PreserveStorage > >	1494
Rcpp::Matrix< RTYPE, StoragePolicy >	1462
Rcpp::MatrixBase< RTYPE, NA, T >	1494
Rcpp::VectorBase< RTYPE, true, Times_Vector_Primitive< RTYPE, NA, T > >	2561



Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T > . . . . .	2384
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 > > . . . . .	2561
Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 > . . . . .	2326
Rcpp::VectorBase< RTYPE, true, Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > . . . . .	1607
Rcpp::VectorBase< RTYPE, true, Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > . . . . .	1611
Rcpp::VectorBase< RTYPE,(COND_NA  LHS_NA  RHS_NA), IfElse< RTYPE, COND_NA, COND_← T, LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	1206
Rcpp::VectorBase< LGLSXP, true, Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T > . . . . .	675
Rcpp::VectorBase< REALSXP, NA, P1< RTYPE, NA, T > > . . . . .	2561
Rcpp::stats::P1< RTYPE, NA, T > . . . . .	1762
Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage > > . . . . .	2561
Rcpp::Vector< STRSXP > . . . . .	2494
Rcpp::Vector< VECSXP > . . . . .	2494
Rcpp::Vector< INTSXP > . . . . .	2494
Rcpp::Vector< RTYPE, PreserveStorage > . . . . .	2494
Rcpp::Vector< RTYPE, StoragePolicy > . . . . .	2494
Rcpp::VectorBase< REALSXP, false, Times_Vector_Primitive< REALSXP, false, T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T > . . . . .	2388
Rcpp::VectorBase< RTYPE, NA, Rep< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Rep< RTYPE, NA, T > . . . . .	2032
Rcpp::VectorBase< RTYPE, true, Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T > > . . . . .	2561
Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T > . . . . .	1862
Rcpp::VectorBase< RTYPE, false, Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > . . . . .	1828
Rcpp::VectorBase< RTYPE, false, Nona< RTYPE, NA, VECTOR > > . . . . .	2561
Rcpp::sugar::Nona< RTYPE, NA, VECTOR > . . . . .	1672
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > . . . . .	1042
Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	2415
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_1< NA, RESULT_TYPE, U1, T1 > > . . . . .	2561
Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 > . . . . .	2314
Rcpp::VectorBase< RTYPE, NA, Rep_each< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Rep_each< RTYPE, NA, T > . . . . .	2035
Rcpp::VectorBase< RTYPE,(LHS_NA  RHS_NA), Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	1851
Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > . . . . .	1727
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	1600
Rcpp::VectorBase< REALSXP, true, Divides_Primitive_Vector< REALSXP, false, T > > . . . . .	2561
Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T > . . . . .	1003

Rcpp::VectorBase< REALSXP, true, Divides_Primitive_Vector< REALSXP, NA, T > > . . . . .	2561
Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T > . . . . .	1007
Rcpp::VectorBase< REALSXP, NA, P3< RTYPE, NA, T > > . . . . .	2561
Rcpp::stats::P3< RTYPE, NA, T > . . . . .	1771
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< T >::rtype, true, Rep_Single< T > > . . . . .	2561
Rcpp::sugar::Rep_Single< T > . . . . .	2042
Rcpp::VectorBase< REALSXP, NA, Pow< RTYPE, NA, T, EXPONENT_TYPE > > . . . . .	2561
Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE > . . . . .	1871
Rcpp::VectorBase< RTYPE, true, Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T > > . . . . .	2561
Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T > . . . . .	1847
Rcpp::VectorBase< RTYPE, true, IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T > > . . . . .	2561
Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T > . . . . .	1231
Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T > . . . . .	1227
Rcpp::VectorBase< RTYPE, true, Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > . . . . .	1053
Rcpp::VectorBase< RTYPE, true, Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > . . . . .	1057
Rcpp::VectorBase< RTYPE, true, MatrixRow< RTYPE > > . . . . .	2561
Rcpp::MatrixRow< RTYPE > . . . . .	1507
Rcpp::VectorBase< REALSXP, false, Vectorized_INTSXP< Func, false, VEC > > . . . . .	2561
Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC > . . . . .	2574
Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > . . . . .	2437
Rcpp::VectorBase< RTYPE, true, Minus_Vector_Primitive< RTYPE, false, T > > . . . . .	2561
Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T > . . . . .	1581
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	1046
Rcpp::VectorBase< RTYPE, true, Divides_Vector_Primitive< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T > . . . . .	1015
Rcpp::VectorBase< LGLSXP, true, And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T > . . . . .	507
Rcpp::VectorBase< INTSXP, false, Range > . . . . .	2561
Rcpp::Range . . . . .	2008
Rcpp::VectorBase< RTYPE, true, ConstMatrixRow< RTYPE > > . . . . .	2561
Rcpp::ConstMatrixRow< RTYPE > . . . . .	760
Rcpp::VectorBase< RTYPE, false, Times_Vector_Primitive< RTYPE, false, T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T > . . . . .	2396
Rcpp::VectorBase< RTYPE, NA, Rep_len< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Rep_len< RTYPE, NA, T > . . . . .	2039
Rcpp::VectorBase< REALSXP, false, Pow< INTSXP, false, T, EXPONENT_TYPE > > . . . . .	2561
Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE > . . . . .	1875
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Primitive< REALSXP, NA, T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T > . . . . .	1787
Rcpp::VectorBase< RTYPE, true, Minus_Vector_Primitive< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T > . . . . .	1569
Rcpp::VectorBase< REALSXP, true, Minus_Primitive_Vector< REALSXP, NA, T > > . . . . .	2561
Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T > . . . . .	1561

Rcpp::VectorBase< LGLSXP, false, IsNaN< RTYPE, NA, VEC_TYPE > > . . . . .	2561
Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE > . . . . .	1343
Rcpp::VectorBase< RTYPE, true, Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	1031
Rcpp::VectorBase< RTYPE, NA, Diag_Extractor< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Diag_Extractor< RTYPE, NA, T > . . . . .	970
Rcpp::VectorBase< RTYPE, NA, RangeIndexer< RTYPE, NA, VECTOR > > . . . . .	2561
Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR > . . . . .	2019
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T > . . . . .	1832
Rcpp::VectorBase< REALSXP, false, Plus_Vector_Primitive< REALSXP, false, T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T > . . . . .	1783
Rcpp::VectorBase< RTYPE, NA, Head< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Head< RTYPE, NA, T > . . . . .	1198
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Primitive< REALSXP, NA, T > > . . . . .	2561
Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T > . . . . .	1023
Rcpp::VectorBase< REALSXP, NA, Vectorized_INTSXP< Func, NA, VEC > > . . . . .	2561
Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC > . . . . .	2571
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, false, SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr > > . . . . .	2561
Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr > . . . . .	2344
Rcpp::VectorBase< INTSXP, NA, Sign< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Sign< RTYPE, NA, T > . . . . .	2154
Rcpp::VectorBase< RTYPE, NA, Rev< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Rev< RTYPE, NA, T > . . . . .	2052
Rcpp::VectorBase< RTYPE, false, Plus_Vector_Primitive_nona< RTYPE, false, T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T > . . . . .	1806
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Primitive_nona< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T > . . . . .	1795
Rcpp::VectorBase< RTYPE, true, Divides_Vector_Primitive< RTYPE, false, T > > . . . . .	2561
Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T > . . . . .	1027
Rcpp::VectorBase< RTYPE, true, Divides_Primitive_Vector< RTYPE, false, T > > . . . . .	2561
Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T > . . . . .	1011
Rcpp::VectorBase< RTYPE, true, Divides_Primitive_Vector< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T > . . . . .	999
Rcpp::VectorBase< RTYPE, false, Diff< RTYPE, false, LHS_T > > . . . . .	2561
Rcpp::sugar::Diff< RTYPE, false, LHS_T > . . . . .	986
Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	2429
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > . . . . .	1821
Rcpp::VectorBase< RTYPE, false, Plus_Vector_Primitive< RTYPE, false, T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T > . . . . .	1791
Rcpp::VectorBase< LGLSXP, false, Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > . . . . .	1724
Rcpp::VectorBase< RTYPE, false, Nona< RTYPE, NA, Rcpp::Vector< RTYPE > > > . . . . .	2561
Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > > . . . . .	1676
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::traits::result_of< Function >::type >::rtype, true, Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function > > . . . . .	2561

Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function > . . . . .	1453
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::traits::result_of< Function >::type >::rtype, true, Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function > > . . . . .	2561
Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function > . . . . .	1447
Rcpp::VectorBase< VECSXP, true, Lapply< RTYPE, NA, T, Function > > . . . . .	2561
Rcpp::sugar::Lapply< RTYPE, NA, T, Function > . . . . .	1397
Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T > . . . . .	1223
Rcpp::VectorBase< LGLSXP, true, And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > . . . . .	510
Rcpp::VectorBase< LGLSXP, true, And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_←_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	500
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::sugar::sapply_application_result_of< Function, T >::type >::rtype, true, Sapply< RTYPE, NA, T, Function, true > > . . . . .	2561
Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true > . . . . .	2116
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::sugar::sapply_application_result_of< Function, T >::type >::rtype, true, Sapply< RTYPE, NA, T, Function, NO_CONVERSION > > . . . . .	2561
Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION > . . . . .	2112
Rcpp::VectorBase< RTYPE,(LHS_NA  RHS_NA), Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_←_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	1865
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > . . . . .	1836
Rcpp::VectorBase< LGLSXP, false, IsNa_Vector_is_na< T > > . . . . .	2561
Rcpp::sugar::IsNa_Vector_is_na< T > . . . . .	1340
Rcpp::VectorBase< RTYPE, true, ConstMatrixColumn< RTYPE > > . . . . .	2561
Rcpp::ConstMatrixColumn< RTYPE > . . . . .	754
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 > > . . . . .	2561
Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 > . . . . .	2317
Rcpp::VectorBase< REALSXP, false, Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > . . . . .	1814
Rcpp::VectorBase< REALSXP, false, Plus_Vector_Primitive_nona< REALSXP, false, T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T > . . . . .	1799
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Primitive_nona< REALSXP, NA, T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T > . . . . .	1802
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::traits::result_of< Function >::type >::rtype, true, Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function > > . . . . .	2561
Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function > . . . . .	1457
Rcpp::VectorBase< LGLSXP, false, IsNa< RTYPE, false, VEC_TYPE > > . . . . .	2561
Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE > . . . . .	1337
Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T > > . . . . .	2561
Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T > . . . . .	1215
Rcpp::VectorBase< RTYPE, LHS_NA, Diff< RTYPE, LHS_NA, LHS_T > > . . . . .	2561
Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T > . . . . .	978
Rcpp::VectorBase< RTYPE, NA, Clamp_Primitive_Vector_Primitive< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T > . . . . .	614

Rcpp::VectorBase< RTYPE, true, MatrixColumn< RTYPE > > . . . . .	2561
Rcpp::MatrixColumn< RTYPE > . . . . .	1499
Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > . . . . .	2441
Rcpp::VectorBase< RTYPE, false, Times_Vector_Primitive_nona< RTYPE, false, T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T > . . . . .	2411
Rcpp::VectorBase< REALSXP, true, Times_Vector_Primitive_nona< REALSXP, NA, T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T > . . . . .	2407
Rcpp::VectorBase< RTYPE, true, Times_Vector_Primitive_nona< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T > . . . . .	2400
Rcpp::VectorBase< REALSXP, NA, Pow< INTSXP, NA, T, EXPONENT_TYPE > > . . . . .	2561
Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE > . . . . .	1878
Rcpp::VectorBase< REALSXP, LHS_NA, Diff< REALSXP, LHS_NA, LHS_T > > . . . . .	2561
Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T > . . . . .	983
Rcpp::VectorBase< LGLSXP, true, Comparator_With_One_Value< RTYPE, Operator, NA, T > > . . . . .	2561
Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T > . . . . .	683
Rcpp::VectorBase< REALSXP, NA, Q0< RTYPE, NA, T > > . . . . .	2561
Rcpp::stats::Q0< RTYPE, NA, T > . . . . .	1897
Rcpp::VectorBase< REALSXP, NA, D0< RTYPE, NA, T > > . . . . .	2561
Rcpp::stats::D0< RTYPE, NA, T > . . . . .	919
Rcpp::VectorBase< REALSXP, NA, Vectorized< Func, NA, VEC > > . . . . .	2561
Rcpp::sugar::Vectorized< Func, NA, VEC > . . . . .	2567
Rcpp::VectorBase< REALSXP, false, Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > . . . . .	2419
Rcpp::VectorBase< REALSXP, false, Times_Vector_Primitive_nona< REALSXP, false, T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T > . . . . .	2404
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarComplex< NA, RESULT_TYPE, T, FunPtr > > . . . . .	2561
Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr > . . . . .	2330
Rcpp::VectorBase< INTSXP, false, SeqLen > . . . . .	2561
Rcpp::sugar::SeqLen . . . . .	2140
Rcpp::VectorBase< RTYPE, false, Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > . . . . .	1603
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > . . . . .	1593
Rcpp::VectorBase< RTYPE, true, IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T > > . . . . .	2561
Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T > . . . . .	1236
Rcpp::VectorBase< RTYPE,(LHS_NA  RHS_NA), IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	1211
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > . . . . .	1039
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Primitive< REALSXP, false, T > > . . . . .	2561
Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T > . . . . .	1019
Rcpp::VectorBase< REALSXP, true, Times_Vector_Primitive< REALSXP, NA, T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T > . . . . .	2392
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr > > . . . . .	2561
Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr > . . . . .	2347



Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 > > . . . . .	2561
Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 > . . . . .	2323
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > . . . . .	1596
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Primitive< REALSXP, NA, T > > . . . . .	2561
Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T > . . . . .	1577
Rcpp::VectorBase< RTYPE, true, Minus_Primitive_Vector< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T > . . . . .	1553
Rcpp::VectorBase< REALSXP, NA, Q2< RTYPE, NA, T > > . . . . .	2561
Rcpp::stats::Q2< RTYPE, NA, T > . . . . .	1905
Rcpp::VectorBase< REALSXP, NA, D2< RTYPE, NA, T > > . . . . .	2561
Rcpp::stats::D2< RTYPE, NA, T > . . . . .	927
Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > . . . . .	2422
Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > . . . . .	2426
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr > > . . . . .	2561
Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr > . . . . .	2341
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Primitive< REALSXP, false, T > > . . . . .	2561
Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T > . . . . .	1573
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::traits::result_of< Function >::type >::rtype, true, Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function > > . . . . .	2561
Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function > . . . . .	1450
Rcpp::VectorBase< LGLSXP, false, IsNa< RTYPE, NA, VEC_TYPE > > . . . . .	2561
Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE > . . . . .	1334
Rcpp::VectorBase< LGLSXP, false, IsInfinite< RTYPE, NA, VEC_TYPE > > . . . . .	2561
Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE > . . . . .	1331
Rcpp::VectorBase< LGLSXP, false, IsFinite< RTYPE, NA, VEC_TYPE > > . . . . .	2561
Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE > . . . . .	1328
Rcpp::VectorBase< RTYPE, false, Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > . . . . .	1049
Rcpp::VectorBase< LGLSXP, true, Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T > . . . . .	679
Rcpp::VectorBase< LGLSXP, true, Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	671
Rcpp::VectorBase< REALSXP, NA, Q1< RTYPE, NA, T > > . . . . .	2561
Rcpp::stats::Q1< RTYPE, NA, T > . . . . .	1901
Rcpp::VectorBase< REALSXP, NA, P0< RTYPE, NA, T > > . . . . .	2561
Rcpp::stats::P0< RTYPE, NA, T > . . . . .	1758
Rcpp::VectorBase< REALSXP, NA, D1< RTYPE, NA, T > > . . . . .	2561
Rcpp::stats::D1< RTYPE, NA, T > . . . . .	923
Rcpp::VectorBase< RTYPE, NA, Tail< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Tail< RTYPE, NA, T > . . . . .	2369
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2561

Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	1824
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Primitive< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T > . . . . .	1779
Rcpp::VectorBase< LGLSXP, NA, Not_Vector< RTYPE, NA, T > > . . . . .	2561
Rcpp::sugar::Not_Vector< RTYPE, NA, T > . . . . .	1700
Rcpp::VectorBase< REALSXP, true, Minus_Primitive_Vector< REALSXP, false, T > > . . . . .	2561
Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T > . . . . .	1557
Rcpp::VectorBase< RTYPE, true, Comparator_With_One_Value< LGLSXP, Operator, false, T > > . . . . .	2561
Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T > . . . . .	688
Rcpp::VectorBase< LGLSXP, false, And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > . . . . .	504
Rcpp::VectorBase< unary_minus_result_type< RTYPE >::value, NA, UnaryMinus_Vector< unary_← minus_result_type< RTYPE >::value, NA, T > > . . . . .	2561
Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T > . . . . .	2465
Rcpp::VectorBase< RTYPE, false, Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > . . . . .	2433
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_2_PV< NA, RESULT_TYPE, U1, U2, T2 > > . . . . .	2561
Rcpp::sugar::SugarBlock_2_PV< NA, RESULT_TYPE, U1, U2, T2 > . . . . .	2320
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > . . . . .	1817
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	1810
Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > . . . . .	1730
Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_← _NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	1720
Rcpp::VectorBase< REALSXP, false, Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > . . . . .	1589
Rcpp::VectorBase< RTYPE, true, Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2561
Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	1585
Rcpp::VectorBase< RTYPE, true, Minus_Primitive_Vector< RTYPE, false, T > > . . . . .	2561
Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T > . . . . .	1565
Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Primitive< RTYPE, false, COND_T > > . . . . .	2561
Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T > . . . . .	1219
Rcpp::VectorBase< REALSXP, false, Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > > . . . . .	2561
Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > . . . . .	1035
Rcpp::VectorBase< REALSXP, NA, Q3< RTYPE, NA, T > > . . . . .	2561
Rcpp::stats::Q3< RTYPE, NA, T > . . . . .	1910
Rcpp::VectorBase< REALSXP, NA, P2< RTYPE, NA, T > > . . . . .	2561
Rcpp::stats::P2< RTYPE, NA, T > . . . . .	1766
Rcpp::VectorBase< REALSXP, NA, D3< RTYPE, NA, T > > . . . . .	2561
Rcpp::stats::D3< RTYPE, NA, T > . . . . .	931
Rcpp::MatrixBase< RTYPE, na, MATRIX > . . . . .	1494
Rcpp::VectorBase< RTYPE, na, VECTOR > . . . . .	2561
Rcpp::Vector< VECSXP, StoragePolicy > . . . . .	2494
Rcpp::traits::expands_to_logical_impl< LGLSXP > . . . . .	1090
Rcpp::attributes::ExportsGenerator . . . . .	1095

Rcpp::attributes::CppExportsGenerator	784
Rcpp::attributes::CppExportsIncludeGenerator	791
Rcpp::attributes::CppPackageIncludeGenerator	824
Rcpp::attributes::REExportsGenerator	2055
Rcpp::attributes::ExportsGenerators	1106
Rcpp::traits::Extractor< RTYPE, NA, VECTOR >	1110
tinyformat::detail::is_convertible< T1, T2 >::fail	1110
Rcpp::Fast< VECTOR >	1111
Rcpp::FieldProxyPolicy< CLASS >	1126
Rcpp::attributes::FileInfo	1134
Rcpp::fixed_call< RESULT_TYPE >	1138
Rcpp::sugar::forbidden_conversion< bool >	1140
Rcpp::sugar::forbidden_conversion< true >	1141
Rcpp::sugar::forbidden_conversion< x >	1140
Rcpp::sugar::conversion_to_bool_is_forbidden< x >	776
tinyformat::detail::FormatArg	1141
tinyformat::FormatList	1145
tinyformat::detail::FormatListN< N >	1147
tinyformat::detail::FormatListN< 0 >	1149
tinyformat::detail::formatValueAsType< T, fmtT, convertible >	1151
tinyformat::detail::formatValueAsType< T, fmtT, true >	1151
Rcpp::attributes::Function	1152
Rcpp::Generator< T >	1163
Rcpp::Generator< double >	1163
Rcpp::stats::BetaGenerator	564
Rcpp::stats::BinomGenerator	580
Rcpp::stats::CauchyGenerator	592
Rcpp::stats::CauchyGenerator_0	594
Rcpp::stats::CauchyGenerator_1	596
Rcpp::stats::ChisqGenerator	608
Rcpp::stats::ExpGenerator	1090
Rcpp::stats::ExpGenerator__rate1	1093
Rcpp::stats::FGenerator_Finite_Finite	1114
Rcpp::stats::FGenerator_Finite_NotFinite	1116
Rcpp::stats::FGenerator_NotFinite_Finite	1119
Rcpp::stats::GammaGenerator	1161
Rcpp::stats::GeomGenerator	1184
Rcpp::stats::HyperGenerator	1201
Rcpp::stats::LNormGenerator	1417
Rcpp::stats::LNormGenerator_0	1419
Rcpp::stats::LNormGenerator_1	1421
Rcpp::stats::LogisGenerator	1425
Rcpp::stats::LogisGenerator_0	1427
Rcpp::stats::LogisGenerator_1	1429
Rcpp::stats::NBinomGenerator	1642
Rcpp::stats::NBinomGenerator_Mu	1645
Rcpp::stats::NChisqGenerator	1647
Rcpp::stats::NormGenerator	1685
Rcpp::stats::NormGenerator__mean0	1687
Rcpp::stats::NormGenerator__mean0__sd1	1690
Rcpp::stats::NormGenerator__sd1	1692
Rcpp::stats::PoissonGenerator	1869
Rcpp::stats::SignRankGenerator	2170



Rcpp::stats::TGenerator . . . . .	2372
Rcpp::stats::UnifGenerator . . . . .	2469
Rcpp::stats::UnifGenerator__0__1 . . . . .	2472
Rcpp::stats::WeibullGenerator . . . . .	2578
Rcpp::stats::WeibullGenerator__scale1 . . . . .	2580
Rcpp::stats::WilcoxGenerator . . . . .	2595
Rcpp::internal::generic_element_converter< RTYPE > . . . . .	1164
Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy > . . . . .	1167
Rcpp::GenericProxy< Proxy > . . . . .	1183
Rcpp::GenericProxy< AttributeProxy > . . . . .	1183
Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy . . . . .	557
Rcpp::GenericProxy< Binding > . . . . .	1183
Rcpp::BindingPolicy< EnvironmentClass >::Binding . . . . .	573
Rcpp::GenericProxy< const_AttributeProxy > . . . . .	1183
Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy . . . . .	694
Rcpp::GenericProxy< const_Binding > . . . . .	1183
Rcpp::BindingPolicy< EnvironmentClass >::const_Binding . . . . .	697
Rcpp::GenericProxy< const_DottedPairProxy > . . . . .	1183
Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy . . . . .	701
Rcpp::GenericProxy< const_FieldProxy > . . . . .	1183
Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy . . . . .	705
Rcpp::GenericProxy< const_generic_proxy< RTYPE, StoragePolicy > > . . . . .	1183
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy > . . . . .	709
Rcpp::GenericProxy< const_NamesProxy > . . . . .	1183
Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy . . . . .	727
Rcpp::GenericProxy< const_ProtectedProxy > . . . . .	1183
Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy . . . . .	730
Rcpp::GenericProxy< const_SlotProxy > . . . . .	1183
Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy . . . . .	733
Rcpp::GenericProxy< const_TagProxy > . . . . .	1183
Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy . . . . .	749
Rcpp::GenericProxy< DottedPairProxy > . . . . .	1183
Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy . . . . .	1066
Rcpp::GenericProxy< FieldProxy > . . . . .	1183
Rcpp::FieldProxyPolicy< CLASS >::FieldProxy . . . . .	1121
Rcpp::GenericProxy< generic_proxy< RTYPE, StoragePolicy > > . . . . .	1183
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy > . . . . .	1173
Rcpp::GenericProxy< NamesProxy > . . . . .	1183
Rcpp::NamesProxyPolicy< CLASS >::NamesProxy . . . . .	1636
Rcpp::GenericProxy< ProtectedProxy > . . . . .	1183
Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy . . . . .	1885
Rcpp::GenericProxy< SlotProxy > . . . . .	1183
Rcpp::SlotProxyPolicy< CLASS >::SlotProxy . . . . .	2188
Rcpp::GenericProxy< TagProxy > . . . . .	1183
Rcpp::TagProxyPolicy< XPtrClass >::TagProxy . . . . .	2364
Rcpp::sugar::Grabber< HASH, RTYPE > . . . . .	1186
GreedyVector	
Rcpp::oldDateVector . . . . .	1717

Rcpp::oldDatetimeVector	1715
Rcpp::sugar::cbind_impl::detail::has_stored_type< T >	1196
Rcpp::traits::identity< T >	1203
Rcpp::traits::if_< COND, LHS, RHS >	1204
Rcpp::traits::if_< false, LHS, RHS >	1205
Rcpp::sugar::ln< RTYPE, TABLE_T >	1240
Rcpp::sugar::IndexHash< RTYPE >	1242
Rcpp::traits::init_type< RTYPE >	1253
Rcpp::traits::init_type< LGLSXP >	1253
Rcpp::traits::init_type< STRSXP >	1254
Rcpp::traits::input_parameter< T >	1255
Rcpp::traits::input_parameter< const T & >	1255
Rcpp::traits::input_parameter< const T >	1256
Rcpp::traits::input_parameter< T & >	1257
Rcpp::InputParameter< T >	1258
Rcpp::sugar::lnSet< HASH >	1259
Rcpp::traits::int2type< I >	1261
Rcpp::traits::integral_constant< _T, _V >	1262
Rcpp::traits::same_type< r_type_traits< T >::r_category, r_type_primitive_tag >	2109
Rcpp::traits::is_primitive< T >	1310
Rcpp::traits::same_type< T::value_type, wchar_t >	2109
Rcpp::traits::is_wide_string< T >	1322
Rcpp::Matrix< RTYPE, StoragePolicy >::can_have_na	588
Rcpp::sugar::is_sugar_vector< T >	1315
Rcpp::sugar::is_sugar_vector< Rcpp::Vector< RTYPE > >	1316
Rcpp::traits::has_na< RTYPE >	1190
Rcpp::traits::has_na< CPLXSXP >	1191
Rcpp::traits::has_na< INTSXP >	1192
Rcpp::traits::has_na< LGLSXP >	1193
Rcpp::traits::has_na< REALSXP >	1194
Rcpp::traits::has_na< STRSXP >	1195
Rcpp::traits::is_arithmetic< typename >	1270
Rcpp::traits::is_arithmetic< const double >	1271
Rcpp::traits::is_arithmetic< const float >	1272
Rcpp::traits::is_arithmetic< const int >	1273
Rcpp::traits::is_arithmetic< const long >	1274
Rcpp::traits::is_arithmetic< const long double >	1275
Rcpp::traits::is_arithmetic< const short >	1276
Rcpp::traits::is_arithmetic< const unsigned int >	1277
Rcpp::traits::is_arithmetic< const unsigned long >	1278
Rcpp::traits::is_arithmetic< const unsigned short >	1279
Rcpp::traits::is_arithmetic< double >	1280
Rcpp::traits::is_arithmetic< float >	1281
Rcpp::traits::is_arithmetic< int >	1282
Rcpp::traits::is_arithmetic< long >	1283
Rcpp::traits::is_arithmetic< long double >	1284
Rcpp::traits::is_arithmetic< short >	1285
Rcpp::traits::is_arithmetic< unsigned int >	1286
Rcpp::traits::is_arithmetic< unsigned long >	1287
Rcpp::traits::is_arithmetic< unsigned short >	1288
Rcpp::traits::is_bool< typename >	1289
Rcpp::traits::is_bool< bool >	1290
Rcpp::traits::is_bool< const bool >	1291
Rcpp::traits::is_bool< volatile bool >	1292

Rcpp::traits::is_const< typename > . . . . .	1293
Rcpp::traits::is_const< _Tp const > . . . . .	1294
Rcpp::traits::is_named< T > . . . . .	1305
Rcpp::traits::is_named< Rcpp::Argument > . . . . .	1307
Rcpp::traits::is_named< named_object< T > > . . . . .	1306
Rcpp::traits::is_pointer< T > . . . . .	1308
Rcpp::traits::is_pointer< T * > . . . . .	1309
Rcpp::traits::is_reference< typename > . . . . .	1311
Rcpp::traits::is_reference< _Tp & > . . . . .	1313
Rcpp::traits::is_trivial< RTYPE > . . . . .	1317
Rcpp::traits::is_trivial< EXPRXP > . . . . .	1318
Rcpp::traits::is_trivial< VECSXP > . . . . .	1319
Rcpp::traits::is_wide_string< char > . . . . .	1324
Rcpp::traits::is_wide_string< const char * > . . . . .	1325
Rcpp::traits::is_wide_string< const wchar_t * > . . . . .	1326
Rcpp::traits::is_wide_string< wchar_t > . . . . .	1327
Rcpp::traits::needs_protection< T > . . . . .	1650
Rcpp::traits::needs_protection< SEXP > . . . . .	1651
Rcpp::traits::r_sexptype_needs_cscast< T > . . . . .	1915
Rcpp::traits::r_sexptype_needs_cscast< Rbyte > . . . . .	1918
Rcpp::traits::r_sexptype_needs_cscast< Rcomplex > . . . . .	1919
Rcpp::traits::r_sexptype_needs_cscast< double > . . . . .	1916
Rcpp::traits::r_sexptype_needs_cscast< int > . . . . .	1917
Rcpp::traits::same_type< T, U > . . . . .	2109
Rcpp::traits::same_type< T, T > . . . . .	2111
Rcpp::traits::integral_constant< bool, _has_iterator_helper< T >::value > . . . . .	1262
Rcpp::traits::has_iterator< T > . . . . .	1189
Rcpp::traits::integral_constant< bool, _has_matrix_interface_helper< T >::value > . . . . .	1262
Rcpp::traits::matrix_interface< T > . . . . .	1486
Rcpp::traits::integral_constant< bool, _has_rtype_helper< T >::value > . . . . .	1262
Rcpp::traits::expands_to_logical< T > . . . . .	1086
Rcpp::traits::integral_constant< bool, _is_eigen_helper< T >::value > . . . . .	1262
Rcpp::traits::is_eigen_base< T > . . . . .	1299
Rcpp::traits::integral_constant< bool, _is_exporter_helper< T >::value > . . . . .	1262
Rcpp::traits::is_exporter< T > . . . . .	1300
Rcpp::traits::integral_constant< bool, _is_generator_helper< T >::value > . . . . .	1262
Rcpp::traits::is_generator< T > . . . . .	1302
Rcpp::traits::integral_constant< bool, _is_importer_helper< T >::value > . . . . .	1262
Rcpp::traits::is_importer< T > . . . . .	1303
Rcpp::traits::integral_constant< bool, _is_sugar_expression_helper< T >::value > . . . . .	1262
Rcpp::traits::is_sugar_expression< T > . . . . .	1314
Rcpp::traits::integral_constant< bool, na > . . . . .	1262
Rcpp::MatrixBase< RTYPE, na, MATRIX >::can_have_na . . . . .	589
Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na . . . . .	590
Rcpp::traits::integral_constant< bool, same_type< r_type_traits< T >::r_category, r_type_module_object_tag >::value  same_type< r_type_traits< T >::r_category, r_type_module_object_pointer_tag >::value  same_type< r_type_traits< T >::r_category, r_type_module_object_const_pointer_tag >::value  same_type< r_type_traits< T >::r_category, r_type_module_object_reference_tag >::value  same_type< r_type_traits< T >::r_category, r_type_module_object_const_reference_tag >::value > . . . . .	1262
Rcpp::traits::is_module_object< T > . . . . .	1304

Rcpp::traits::integral_constant< bool, T::value &&U::value > . . . . .	1262
Rcpp::traits::both< T, U > . . . . .	583
Rcpp::traits::integral_constant< int, RTYPE > . . . . .	1262
Rcpp::Matrix< RTYPE, StoragePolicy >::r_type . . . . .	1936
Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_type . . . . .	1938
Rcpp::VectorBase< RTYPE, na, VECTOR >::r_type . . . . .	1939
Rcpp::internal::InterruptedException . . . . .	1266
Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	1266
Rcpp::traits::is_convertible< T, U > . . . . .	1295
tinyformat::detail::is_convertible< T1, T2 > . . . . .	1297
tinyformat::detail::is_wchar< T > . . . . .	1320
tinyformat::detail::is_wchar< const wchar_t * > . . . . .	1321
tinyformat::detail::is_wchar< const wchar_t[n] > . . . . .	1321
tinyformat::detail::is_wchar< wchar_t * > . . . . .	1322
tinyformat::detail::is_wchar< wchar_t[n] > . . . . .	1322
Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS > . . . . .	1346
Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS > . . . . .	1355
Rcpp::MatrixRow< RTYPE >::iter_traits . . . . .	1363
Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits . . . . .	1365
Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator . . . . .	1367
Rcpp::sugar::Lazy< T, EXPR > . . . . .	1401
Rcpp::sugar::ColMeansImpl< RTYPE, false, T, false > . . . . .	651
Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA_RM > . . . . .	654
Rcpp::sugar::ColSumsImpl< RTYPE, false, T, false > . . . . .	660
Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA_RM > . . . . .	663
Rcpp::sugar::RowMeansImpl< RTYPE, false, T, false > . . . . .	2069
Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA_RM > . . . . .	2072
Rcpp::sugar::RowSumsImpl< RTYPE, false, T, false > . . . . .	2079
Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA_RM > . . . . .	2083
Rcpp::sugar::Lazy< detail::ColMeansReturn< RTYPE >::type, ColMeansImpl< RTYPE, NA, T, false > > . . . . .	1401
Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM > . . . . .	651
Rcpp::sugar::Lazy< detail::ColMeansReturn< RTYPE >::type, ColMeansImpl< RTYPE, NA, T, true > > . . . . .	1401
Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true > . . . . .	655
Rcpp::sugar::Lazy< detail::ColSumsReturn< RTYPE >::type, ColSumsImpl< RTYPE, NA, T, false > > . . . . .	1401
Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM > . . . . .	660
Rcpp::sugar::Lazy< detail::ColSumsReturn< RTYPE >::type, ColSumsImpl< RTYPE, NA, T, true > > . . . . .	1401
Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true > . . . . .	664
Rcpp::sugar::Lazy< detail::RowMeansReturn< RTYPE >::type, RowMeansImpl< RTYPE, NA, T, false > > . . . . .	1401
Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM > . . . . .	2069
Rcpp::sugar::Lazy< detail::RowMeansReturn< RTYPE >::type, RowMeansImpl< RTYPE, NA, T, true > > . . . . .	1401
Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true > . . . . .	2073
Rcpp::sugar::Lazy< detail::RowSumsReturn< RTYPE >::type, RowSumsImpl< RTYPE, NA, T, false > > . . . . .	1401
Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM > . . . . .	2079
Rcpp::sugar::Lazy< detail::RowSumsReturn< RTYPE >::type, RowSumsImpl< RTYPE, NA, T, true > > . . . . .	1401
Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true > . . . . .	2084
Rcpp::sugar::Lazy< double, Mean< INTSXP, NA, T > > . . . . .	1401
Rcpp::sugar::Mean< INTSXP, NA, T > . . . . .	1527
Rcpp::sugar::Lazy< double, Mean< LGLSXP, NA, T > > . . . . .	1401
Rcpp::sugar::Mean< LGLSXP, NA, T > . . . . .	1530
Rcpp::sugar::Lazy< double, Mean< RTYPE, NA, T > > . . . . .	1401

Rcpp::sugar::Mean< RTYPE, NA, T > . . . . .	1520
Rcpp::sugar::Lazy< double, Sum< REALSXP, NA, T > > . . . . .	1401
Rcpp::sugar::Sum< REALSXP, NA, T > . . . . .	2354
Rcpp::sugar::Lazy< double, Var< CPLXSXP, NA, T > > . . . . .	1401
Rcpp::sugar::Var< CPLXSXP, NA, T > . . . . .	2486
Rcpp::sugar::Lazy< double, Var< RTYPE, NA, T > > . . . . .	1401
Rcpp::sugar::Var< RTYPE, NA, T > . . . . .	2483
Rcpp::sugar::Lazy< Rcomplex, Mean< CPLXSXP, NA, T > > . . . . .	1401
Rcpp::sugar::Mean< CPLXSXP, NA, T > . . . . .	1524
Rcpp::sugar::Lazy< Rcpp::traits::storage_type< RTYPE >::type, Sd< RTYPE, NA, T > > . . . . .	1401
Rcpp::sugar::Sd< RTYPE, NA, T > . . . . .	2125
Rcpp::sugar::Lazy< Rcpp::traits::storage_type< RTYPE >::type, Sum< RTYPE, false, T > > . . . . .	1401
Rcpp::sugar::Sum< RTYPE, false, T > . . . . .	2356
Rcpp::sugar::Lazy< Rcpp::traits::storage_type< RTYPE >::type, Sum< RTYPE, NA, T > > . . . . .	1401
Rcpp::sugar::Sum< RTYPE, NA, T > . . . . .	2351
Rcpp::sugar::Lazy< Rcpp::Vector< RTYPE >, Cummax< RTYPE, NA, T > > . . . . .	1401
Rcpp::sugar::Cummax< RTYPE, NA, T > . . . . .	907
Rcpp::sugar::Lazy< Rcpp::Vector< RTYPE >, Cummin< RTYPE, NA, T > > . . . . .	1401
Rcpp::sugar::Cummin< RTYPE, NA, T > . . . . .	910
Rcpp::sugar::Lazy< Rcpp::Vector< RTYPE >, Cumprod< RTYPE, NA, T > > . . . . .	1401
Rcpp::sugar::Cumprod< RTYPE, NA, T > . . . . .	913
Rcpp::sugar::Lazy< Rcpp::Vector< RTYPE >, Cumsum< RTYPE, NA, T > > . . . . .	1401
Rcpp::sugar::Cumsum< RTYPE, NA, T > . . . . .	916
Rcpp::internal::LazyVector< VECTOR > . . . . .	1402
Rcpp::internal::LazyVector< LHS_TYPE > . . . . .	1402
Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > > . . . . .	1405
Rcpp::internal::LazyVector< RHS_TYPE > . . . . .	1402
Rcpp::algorithm::helpers::log . . . . .	1423
Rcpp::LongjumpException . . . . .	1431
Rcpp::linfo . . . . .	1439
Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, is_container > . . . . .	1439
Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true > . . . . .	1440
Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true > . . . . .	1442
Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, false > . . . . .	1443
Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, true > . . . . .	1445
Rcpp::sugar::cbind_impl::detail::matrix_return< T, is_container > . . . . .	1487
Rcpp::sugar::cbind_impl::detail::matrix_return< bool, false > . . . . .	1488
Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Matrix< LGLSXP >, true > . . . . .	1489
Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Vector< LGLSXP >, true > . . . . .	1489
Rcpp::sugar::cbind_impl::detail::matrix_return< T, detail::has_stored_type< T >::value > . . . . .	1487
Rcpp::sugar::cbind_impl::matrix_return< T, B > . . . . .	1487
Rcpp::sugar::cbind_impl::detail::matrix_return< T, false > . . . . .	1490
Rcpp::sugar::cbind_impl::matrix_return< T, false > . . . . .	1491
Rcpp::sugar::cbind_impl::detail::matrix_return< T, true > . . . . .	1493
Rcpp::sugar::Max< RTYPE, NA, T > . . . . .	1516
Rcpp::sugar::Max< RTYPE, false, T > . . . . .	1518
Rcpp::sugar::Median< RTYPE, NA, T, NA_RM > . . . . .	1532
Rcpp::sugar::Median< RTYPE, false, T, NA_RM > . . . . .	1536
Rcpp::sugar::Median< RTYPE, NA, T, true > . . . . .	1538
Rcpp::sugar::Median< STRSXP, false, T, true > . . . . .	1541

Rcpp::sugar::Median< STRSXP, NA, T, NA_RM > . . . . .	1544
Rcpp::sugar::Median< STRSXP, NA, T, true > . . . . .	1547
Rcpp::sugar::Min< RTYPE, NA, T > . . . . .	1549
Rcpp::sugar::Min< RTYPE, false, T > . . . . .	1551
Rcpp::Module . . . . .	1614
Rcpp::traits::module_wrap_traits< T > . . . . .	1624
Rcpp::traits::module_wrap_traits< T * > . . . . .	1625
Rcpp::traits::module_wrap_traits< void > . . . . .	1626
Rcpp::Na_Proxy . . . . .	1626
Rcpp::traits::named_object< T > . . . . .	1631
Rcpp::traits::named_object< SEXP > . . . . .	1632
Rcpp::internal::NamedPlaceholder . . . . .	1635
Rcpp::NamesProxyPolicy< CLASS > . . . . .	1641
Rcpp::Vector< VECSXP, StoragePolicy > . . . . .	2494
Rcpp::NamesProxyPolicy< T > . . . . .	1641
Rcpp::ListOf< T > . . . . .	1408
Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > > . . . . .	1641
Rcpp::Vector< STRSXP > . . . . .	2494
Rcpp::Vector< VECSXP > . . . . .	2494
Rcpp::Vector< INTSXP > . . . . .	2494
Rcpp::Vector< RTYPE, PreserveStorage > . . . . .	2494
Rcpp::Vector< RTYPE, StoragePolicy > . . . . .	2494
Rcpp::sugar::negate< NA > . . . . .	1652
Rcpp::sugar::negate< false > . . . . .	1653
Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::no . . . . .	1666
Rcpp::no_init_matrix . . . . .	1666
Rcpp::no_init_vector . . . . .	1668
Rcpp::sugar::NonaPrimitive< T > . . . . .	1680
Rcpp::NoProtectStorage< CLASS > . . . . .	1681
Rcpp::traits::normal_wrap_tag . . . . .	1684
Rcpp::sugar::not_< RTYPE, NA > . . . . .	1694
Rcpp::sugar::not_< CPLXSXP, false > . . . . .	1695
Rcpp::sugar::not_< CPLXSXP, NA > . . . . .	1696
Rcpp::sugar::not_< REALSXP, false > . . . . .	1697
Rcpp::sugar::not_< REALSXP, NA > . . . . .	1698
Rcpp::sugar::not_< RTYPE, false > . . . . .	1699
Rcpp::Nullable< T > . . . . .	1705
Rcpp::traits::num2type< N > . . . . .	1712
Rcpp::object< T > . . . . .	1712
Rcpp::traits::one_type< T > . . . . .	1718
std::ostream	
Rcpp::Rostream< OUTPUT > . . . . .	2063
Rcpp::attributes::Param . . . . .	1775
Rcpp::sugar::pmax_op< RTYPE, LHS_NA, RHS_NA > . . . . .	1839
Rcpp::sugar::pmax_op< INTSXP, LHS_NA, RHS_NA > . . . . .	1840
Rcpp::sugar::pmax_op< REALSXP, false, false > . . . . .	1840
Rcpp::sugar::pmax_op< REALSXP, false, true > . . . . .	1841
Rcpp::sugar::pmax_op< REALSXP, true, false > . . . . .	1842
Rcpp::sugar::pmax_op< REALSXP, true, true > . . . . .	1843
Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA > . . . . .	1844
Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true > . . . . .	1845
Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, LHS_NA > . . . . .	1844
Rcpp::sugar::pmin_op< RTYPE, LHS_NA, RHS_NA > . . . . .	1854

Rcpp::sugar::pmin_op< INTSXP, LHS_NA, RHS_NA > . . . . .	1854
Rcpp::sugar::pmin_op< REALSXP, false, false > . . . . .	1855
Rcpp::sugar::pmin_op< REALSXP, false, true > . . . . .	1856
Rcpp::sugar::pmin_op< REALSXP, true, false > . . . . .	1857
Rcpp::sugar::pmin_op< REALSXP, true, true > . . . . .	1857
Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA > . . . . .	1858
Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true > . . . . .	1860
Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, LHS_NA > . . . . .	1858
Rcpp::traits::pointer_wrap_tag . . . . .	1868
Rcpp::PreserveStorage< CLASS > . . . . .	1881
Rcpp::PreserveStorage< Vector< RTYPE, PreserveStorage > > . . . . .	1881
Rcpp::Vector< STRSXP > . . . . .	2494
Rcpp::Vector< VECSXP > . . . . .	2494
Rcpp::Vector< INTSXP > . . . . .	2494
Rcpp::Vector< RTYPE, PreserveStorage > . . . . .	2494
Rcpp::Vector< RTYPE, StoragePolicy > . . . . .	2494
Rcpp::PreserveStorage< XPtr< T, PreserveStorage, standard_delete_finalizer< T >, false > > . . . . .	1881
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > . . . . .	2617
Rcpp::ProtectedProxyPolicy< XPtrClass > . . . . .	1889
Rcpp::ProtectedProxyPolicy< XPtr< T, PreserveStorage, standard_delete_finalizer< T >, false > > . . . . .	1889
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > . . . . .	2617
Rcpp::traits::proxy_based_const_iterator< RTYPE, StoragePolicy > . . . . .	1890
Rcpp::traits::proxy_based_const_iterator< EXPRSXP, StoragePolicy > . . . . .	1890
Rcpp::traits::r_vector_const_iterator< EXPRSXP, StoragePolicy > . . . . .	1987
Rcpp::traits::proxy_based_const_iterator< STRSXP, StoragePolicy > . . . . .	1890
Rcpp::traits::r_vector_const_iterator< STRSXP, StoragePolicy > . . . . .	1988
Rcpp::traits::proxy_based_const_iterator< VECSXP, StoragePolicy > . . . . .	1890
Rcpp::traits::r_vector_const_iterator< VECSXP, StoragePolicy > . . . . .	1990
Rcpp::traits::proxy_based_iterator< RTYPE, StoragePolicy > . . . . .	1891
Rcpp::traits::proxy_based_iterator< EXPRSXP, StoragePolicy > . . . . .	1891
Rcpp::traits::r_vector_iterator< EXPRSXP, StoragePolicy > . . . . .	1998
Rcpp::traits::proxy_based_iterator< STRSXP, StoragePolicy > . . . . .	1891
Rcpp::traits::r_vector_iterator< STRSXP, StoragePolicy > . . . . .	1999
Rcpp::traits::proxy_based_iterator< VECSXP, StoragePolicy > . . . . .	1891
Rcpp::traits::r_vector_iterator< VECSXP, StoragePolicy > . . . . .	2000
Rcpp::traits::proxy_cache< RTYPE, StoragePolicy > . . . . .	1892
Rcpp::traits::expands_to_logical_impl< LGLSXP >::r_expands_to_logical . . . . .	1914
Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_matrix_interface . . . . .	1915
Rcpp::traits::r_sexptype_traits< T > . . . . .	1920
Rcpp::sugar::cbind_impl::cbind_sexptype_traits< T > . . . . .	599
Rcpp::traits::r_sexptype_traits< bool > . . . . .	1921
Rcpp::traits::r_sexptype_traits< const double > . . . . .	1922
Rcpp::traits::r_sexptype_traits< const int > . . . . .	1923
Rcpp::traits::r_sexptype_traits< double > . . . . .	1924
Rcpp::traits::r_sexptype_traits< float > . . . . .	1924
Rcpp::traits::r_sexptype_traits< int > . . . . .	1925
Rcpp::traits::r_sexptype_traits< long > . . . . .	1926
Rcpp::traits::r_sexptype_traits< long double > . . . . .	1927
Rcpp::traits::r_sexptype_traits< Rbyte > . . . . .	1927
Rcpp::traits::r_sexptype_traits< Rcomplex > . . . . .	1928
Rcpp::traits::r_sexptype_traits< Rcpp::Date > . . . . .	1929



Rcpp::traits::r_sexptype_traits< Rcpp::Datetime > . . . . .	1930
Rcpp::traits::r_sexptype_traits< Rcpp::String > . . . . .	1930
Rcpp::traits::r_sexptype_traits< short > . . . . .	1931
Rcpp::traits::r_sexptype_traits< std::complex< double > > . . . . .	1932
Rcpp::traits::r_sexptype_traits< std::complex< float > > . . . . .	1933
Rcpp::traits::r_sexptype_traits< std::string > . . . . .	1933
Rcpp::traits::r_sexptype_traits< unsigned int > . . . . .	1934
Rcpp::traits::r_sexptype_traits< unsigned long > . . . . .	1935
Rcpp::traits::r_sexptype_traits< unsigned short > . . . . .	1936
Rcpp::traits::r_type_enum_tag . . . . .	1940
Rcpp::traits::r_type_generic_tag . . . . .	1940
Rcpp::traits::r_type_module_object_const_pointer_tag . . . . .	1940
Rcpp::traits::r_type_module_object_const_reference_tag . . . . .	1941
Rcpp::traits::r_type_module_object_pointer_tag . . . . .	1941
Rcpp::traits::r_type_module_object_reference_tag . . . . .	1941
Rcpp::traits::r_type_module_object_tag . . . . .	1942
Rcpp::traits::r_type_pair_tag . . . . .	1942
Rcpp::traits::r_type_pairstring_generic_tag . . . . .	1942
Rcpp::traits::r_type_pairstring_primitive_tag . . . . .	1943
Rcpp::traits::r_type_pairstring_string_tag . . . . .	1943
Rcpp::traits::r_type_primitive_tag . . . . .	1943
Rcpp::traits::r_type_RcppString_tag . . . . .	1944
Rcpp::traits::r_type_string_tag . . . . .	1944
Rcpp::traits::r_type_traits< T > . . . . .	1944
Rcpp::traits::r_type_traits< bool > . . . . .	1945
Rcpp::traits::r_type_traits< char > . . . . .	1946
Rcpp::traits::r_type_traits< const char * > . . . . .	1946
Rcpp::traits::r_type_traits< const double > . . . . .	1947
Rcpp::traits::r_type_traits< const int > . . . . .	1948
Rcpp::traits::r_type_traits< const wchar_t * > . . . . .	1948
Rcpp::traits::r_type_traits< double > . . . . .	1949
Rcpp::traits::r_type_traits< float > . . . . .	1950
Rcpp::traits::r_type_traits< int > . . . . .	1950
Rcpp::traits::r_type_traits< long > . . . . .	1951
Rcpp::traits::r_type_traits< long double > . . . . .	1952
Rcpp::traits::r_type_traits< Rbyte > . . . . .	1952
Rcpp::traits::r_type_traits< Rcomplex > . . . . .	1953
Rcpp::traits::r_type_traits< Rcpp::Date > . . . . .	1954
Rcpp::traits::r_type_traits< Rcpp::Datetime > . . . . .	1954
Rcpp::traits::r_type_traits< Rcpp::object< T > > . . . . .	1955
Rcpp::traits::r_type_traits< Rcpp::String > . . . . .	1956
Rcpp::traits::r_type_traits< short > . . . . .	1956
Rcpp::traits::r_type_traits< std::complex< double > > . . . . .	1957
Rcpp::traits::r_type_traits< std::complex< float > > . . . . .	1958
Rcpp::traits::r_type_traits< std::pair< const KEY, VALUE > > . . . . .	1958
Rcpp::traits::r_type_traits< std::pair< const std::string, bool > > . . . . .	1959
Rcpp::traits::r_type_traits< std::pair< const std::string, char > > . . . . .	1960
Rcpp::traits::r_type_traits< std::pair< const std::string, const int > > . . . . .	1961
Rcpp::traits::r_type_traits< std::pair< const std::string, double > > . . . . .	1961
Rcpp::traits::r_type_traits< std::pair< const std::string, float > > . . . . .	1962
Rcpp::traits::r_type_traits< std::pair< const std::string, int > > . . . . .	1963
Rcpp::traits::r_type_traits< std::pair< const std::string, long > > . . . . .	1963
Rcpp::traits::r_type_traits< std::pair< const std::string, long double > > . . . . .	1964
Rcpp::traits::r_type_traits< std::pair< const std::string, Rbyte > > . . . . .	1965



Rcpp::traits::r_type_traits< std::pair< const std::string, Rcomplex > > . . . . .	1965
Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Date > > . . . . .	1966
Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Datetime > > . . . . .	1967
Rcpp::traits::r_type_traits< std::pair< const std::string, short > > . . . . .	1967
Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex< double > > > . . . . .	1968
Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex< float > > > . . . . .	1969
Rcpp::traits::r_type_traits< std::pair< const std::string, std::string > > . . . . .	1969
Rcpp::traits::r_type_traits< std::pair< const std::string, std::wstring > > . . . . .	1970
Rcpp::traits::r_type_traits< std::pair< const std::string, T > > . . . . .	1971
Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned int > > . . . . .	1971
Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned long > > . . . . .	1972
Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned short > > . . . . .	1973
Rcpp::traits::r_type_traits< std::pair< const std::string, wchar_t > > . . . . .	1974
Rcpp::traits::r_type_traits< std::string > . . . . .	1974
Rcpp::traits::r_type_traits< std::wstring > . . . . .	1975
Rcpp::traits::r_type_traits< unsigned int > . . . . .	1976
Rcpp::traits::r_type_traits< unsigned long > . . . . .	1976
Rcpp::traits::r_type_traits< unsigned short > . . . . .	1977
Rcpp::traits::r_type_traits< wchar_t > . . . . .	1978
Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy > . . . . .	1978
Rcpp::traits::r_vector_cache_type< RTYPE, StoragePolicy > . . . . .	1983
Rcpp::traits::r_vector_cache_type< EXPRXP, StoragePolicy > . . . . .	1984
Rcpp::traits::r_vector_cache_type< RTYPE, PreserveStorage > . . . . .	1983
Rcpp::traits::r_vector_cache_type< STRSXP, StoragePolicy > . . . . .	1984
Rcpp::traits::r_vector_cache_type< VECSXP, StoragePolicy > . . . . .	1985
Rcpp::traits::r_vector_const_iterator< RTYPE, StoragePolicy > . . . . .	1986
Rcpp::traits::r_vector_const_proxy< RTYPE, StoragePolicy > . . . . .	1991
Rcpp::traits::r_vector_const_proxy< EXPRXP, StoragePolicy > . . . . .	1992
Rcpp::traits::r_vector_const_proxy< STRSXP, StoragePolicy > . . . . .	1992
Rcpp::traits::r_vector_const_proxy< VECSXP, StoragePolicy > . . . . .	1993
Rcpp::traits::r_vector_element_converter< RTYPE > . . . . .	1994
Rcpp::traits::r_vector_element_converter< EXPRXP > . . . . .	1995
Rcpp::traits::r_vector_element_converter< STRSXP > . . . . .	1996
Rcpp::traits::r_vector_element_converter< VECSXP > . . . . .	1996
Rcpp::traits::r_vector_iterator< RTYPE, StoragePolicy > . . . . .	1997
Rcpp::traits::r_vector_iterator< RTYPE > . . . . .	1997
Rcpp::traits::r_vector_name_proxy< RTYPE, StoragePolicy > . . . . .	2001
Rcpp::traits::r_vector_name_proxy< EXPRXP, StoragePolicy > . . . . .	2002
Rcpp::traits::r_vector_name_proxy< STRSXP, StoragePolicy > . . . . .	2003
Rcpp::traits::r_vector_name_proxy< VECSXP, StoragePolicy > . . . . .	2004
Rcpp::traits::r_vector_proxy< RTYPE, StoragePolicy > . . . . .	2005
Rcpp::traits::r_vector_proxy< EXPRXP, StoragePolicy > . . . . .	2005
Rcpp::traits::r_vector_proxy< STRSXP, StoragePolicy > . . . . .	2006
Rcpp::traits::r_vector_proxy< VECSXP, StoragePolicy > . . . . .	2007
Rcpp::sugar::Range< RTYPE, NA, T > . . . . .	2014
Rcpp::sugar::Range< RTYPE, false, T > . . . . .	2016
Rcpp::VectorBase< RTYPE, na, VECTOR >::rcpp_sugar_expression . . . . .	2024
Rcpp::Reference	
Rcpp::S4_CppOverloadedMethods< Class > . . . . .	2103
Rcpp::S4_field< Class > . . . . .	2107
Reference	
Rcpp::S4_CppConstructor< Class > . . . . .	2101
Rcpp::ReferenceInputParameter< T > . . . . .	2024
Rcpp::traits::remove_const< _Tp > . . . . .	2026

Rcpp::traits::remove_const< _Tp const > . . . . .	2027
Rcpp::traits::remove_const_and_reference< T > . . . . .	2028
Rcpp::traits::remove_reference< _Tp > . . . . .	2028
Rcpp::traits::remove_reference< _Tp & > . . . . .	2029
Rcpp::sugar::RemoveFromSet< SET > . . . . .	2030
Rcpp::result< T > . . . . .	2045
Rcpp::sugar::median_detail::result< RTYPE > . . . . .	2046
Rcpp::sugar::median_detail::result< INTSXP > . . . . .	2047
Rcpp::sugar::median_detail::result< STRSXP > . . . . .	2048
Rcpp::traits::result_of< T > . . . . .	2049
Rcpp::traits::result_of< RESULT_TYPE(*) (INPUT_TYPE) > . . . . .	2050
Rcpp::traits::result_of< RESULT_TYPE(*) (U1, U2) > . . . . .	2051
Rcpp::traits::result_of< RESULT_TYPE(*) (U1, U2, U3) > . . . . .	2051
Rcpp::RNGScope . . . . .	2059
Rcpp::ROjectMethods< Class > . . . . .	2061
Rcpp::Vector< VECSXP, StoragePolicy > . . . . .	2494
Rcpp::ROjectMethods< T > . . . . .	2061
Rcpp::ListOf< T > . . . . .	1408
Rcpp::ROjectMethods< Vector< RTYPE, PreserveStorage > > . . . . .	2061
Rcpp::Vector< STRSXP > . . . . .	2494
Rcpp::Vector< VECSXP > . . . . .	2494
Rcpp::Vector< INTSXP > . . . . .	2494
Rcpp::Vector< RTYPE, PreserveStorage > . . . . .	2494
Rcpp::Vector< RTYPE, StoragePolicy > . . . . .	2494
Rcpp::ROjectMethods< XPtr< T, PreserveStorage, standard_delete_finalizer< T >, false > > . . . . .	2061
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > . . . . .	2617
Rcpp::sugar::detail::RowMeansReturn< RTYPE > . . . . .	2077
Rcpp::sugar::detail::ColMeansReturn< RTYPE > . . . . .	659
Rcpp::sugar::detail::RowMeansReturn< CPLXSXP > . . . . .	2078
Rcpp::sugar::detail::RowSumsReturn< RTYPE > . . . . .	2087
Rcpp::sugar::detail::ColSumsReturn< RTYPE > . . . . .	668
Rcpp::sugar::detail::RowSumsReturn< LGLSXP > . . . . .	2089
Rcpp::algorithm::helpers::rtype< T > . . . . .	2093
Rcpp::algorithm::helpers::rtype_helper< T > . . . . .	2095
Rcpp::algorithm::helpers::rtype_helper< double > . . . . .	2096
Rcpp::algorithm::helpers::rtype_helper< int > . . . . .	2097
Rcpp::rule . . . . .	2099
S4 . . . . .	
Rcpp::CppClass . . . . .	781
Rcpp::CppObject . . . . .	822
Rcpp::sugar::sapply_application_result_of< Function, SugarExpression > . . . . .	2120
Rcpp::sugar::cbind_impl::scalar< RTYPE > . . . . .	2121
Rcpp::sugar::SelfHash< RTYPE > . . . . .	2128
Rcpp::sugar::SelfInserter< HASH, STORAGE > . . . . .	2134
Rcpp::sugar::SelfMatch< RTYPE, TABLE_T > . . . . .	2136
Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	2142
Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	2145
Rcpp::Shelter< T > . . . . .	2148
Rcpp::Shield< T > . . . . .	2151
Rcpp::sugar::sign_impl< NA, RTYPE > . . . . .	2158
Rcpp::sugar::sign_impl< false, RTYPE > . . . . .	2159
Rcpp::SignedConstructor< Class > . . . . .	2160

Rcpp::SignedFactory< Class > . . . . .	2163
Rcpp::SignedMethod< Class > . . . . .	2165
Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy > . . . . .	2172
Rcpp::SingleLogicalResult< NA, T > . . . . .	2178
Rcpp::sugar::SingleLogicalResult< NA, T > . . . . .	2179
Rcpp::sugar::SingleLogicalResult< false, All< false, T > > . . . . .	2179
Rcpp::sugar::All< false, T > . . . . .	497
Rcpp::sugar::SingleLogicalResult< false, And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T > > . . . . .	2179
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T > . . . . .	521
Rcpp::sugar::SingleLogicalResult< false, Any< false, T > > . . . . .	2179
Rcpp::sugar::Any< false, T > . . . . .	535
Rcpp::sugar::SingleLogicalResult< false, Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T > > . . . . .	2179
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T > . . . . .	1741
Rcpp::sugar::SingleLogicalResult< LHS_NA, And_SingleLogicalResult_bool< LHS_NA, LHS_T > > . . . . .	2179
Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T > . . . . .	513
Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T > . . . . .	1733
Rcpp::sugar::SingleLogicalResult< LHS_NA, And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > > . . . . .	2179
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > . . . . .	528
Rcpp::sugar::SingleLogicalResult< LHS_NA, Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > > . . . . .	2179
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T > . . . . .	1748
Rcpp::sugar::SingleLogicalResult< NA, Negate_SingleLogicalResult< NA, T > > . . . . .	2179
Rcpp::sugar::Negate_SingleLogicalResult< NA, T > . . . . .	1653
Rcpp::sugar::SingleLogicalResult< RHS_NA, And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > > . . . . .	2179
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > . . . . .	524
Rcpp::sugar::SingleLogicalResult< RHS_NA, Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > > . . . . .	2179
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T > . . . . .	1744
Rcpp::sugar::SingleLogicalResult< true, All< NA, T > > . . . . .	2179
Rcpp::sugar::All< NA, T > . . . . .	493
Rcpp::sugar::SingleLogicalResult< true, Any< NA, T > > . . . . .	2179
Rcpp::sugar::Any< NA, T > . . . . .	531
Rcpp::sugar::SingleLogicalResult<(LHS_NA  RHS_NA), And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2179
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	517
Rcpp::sugar::SingleLogicalResult<(LHS_NA  RHS_NA), Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > > . . . . .	2179
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	1737
Rcpp::SlotProxyPolicy< CLASS > . . . . .	2193
Rcpp::Vector< VECSXP, StoragePolicy > . . . . .	2494
Rcpp::SlotProxyPolicy< Vector< RTYPE, PreserveStorage > > . . . . .	2193
Rcpp::Vector< STRSXP > . . . . .	2494
Rcpp::Vector< VECSXP > . . . . .	2494
Rcpp::Vector< INTSXP > . . . . .	2494
Rcpp::Vector< RTYPE, PreserveStorage > . . . . .	2494

Rcpp::Vector< RTYPE, StoragePolicy > . . . . .	2494
Rcpp::SlotProxyPolicy< XPtr< T, PreserveStorage, standard_delete_finalizer< T >, false > > . . . . .	2193
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > . . . . .	2617
Rcpp::attributes::SourceFileAttributes . . . . .	2194
Rcpp::attributes::SourceFileAttributesParser . . . . .	2198
Rcpp::algorithm::helpers::sqrt . . . . .	2207
Rcpp::state . . . . .	2209
Rcpp::traits::storage_type< RTYPE > . . . . .	2212
Rcpp::sugar::cbind_impl::cbind_storage_type< RTYPE > . . . . .	601
Rcpp::traits::storage_type< CPLXSXP > . . . . .	2213
Rcpp::traits::storage_type< INTSXP > . . . . .	2213
Rcpp::traits::storage_type< LGLSXP > . . . . .	2214
Rcpp::traits::storage_type< RAWSXP > . . . . .	2215
Rcpp::traits::storage_type< REALSXP > . . . . .	2215
StoragePolicy	
Rcpp::Vector< VECSXP, StoragePolicy > . . . . .	2494
std::streambuf	
Rcpp::Rstreambuf< OUTPUT > . . . . .	2090
Rcpp::String . . . . .	2216
Rcpp::internal::string_element_converter< RTYPE > . . . . .	2259
Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy > . . . . .	2261
Rcpp::internal::string_proxy< RTYPE, StoragePolicy > . . . . .	2270
Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T > . . . . .	2299
tinyformat::detail::is_convertible< T1, T2 >::succeed . . . . .	2311
Rcpp::sugar::sugar_const_iterator_type< T > . . . . .	2311
Rcpp::sugar::sugar_const_iterator_type< CharacterVector > . . . . .	2312
Rcpp::sugar::sugar_const_iterator_type< Rcpp::Vector< RTYPE > > . . . . .	2313
Rcpp::sugar::SugarIterator< T > . . . . .	2333
Rcpp::SuspendRNGSynchronizationScope . . . . .	2359
T	
Rcpp::ChildVector< T > . . . . .	605
Rcpp::sugar::Table< RTYPE, TABLE_T > . . . . .	2360
Rcpp::TagProxyPolicy< XPtrClass > . . . . .	2367
Rcpp::TagProxyPolicy< XPtr< T, PreserveStorage, standard_delete_finalizer< T >, false > > . . . . .	2367
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > . . . . .	2617
Rcpp::Timer . . . . .	2374
Timer . . . . .	2378
Rcpp::tinfo . . . . .	2444
Rcpp::attributes::Type . . . . .	2446
T::can_have_na::type	
Rcpp::can_have_na< T > . . . . .	587
Rcpp::tzhead . . . . .	2451
Rcpp::traits::un_pointer< T > . . . . .	2453
Rcpp::traits::un_pointer< object< T > > . . . . .	2454
Rcpp::traits::un_pointer< T * > . . . . .	2455
std::unary_function	
Rcpp::StringTransformer< UnaryOperator > . . . . .	2291
Rcpp::unary_call< T, RESULT_TYPE > . . . . .	2456
Rcpp::sugar::unary_minus< RTYPE, NA > . . . . .	2459
Rcpp::sugar::unary_minus< CPLXSXP, false > . . . . .	2460
Rcpp::sugar::unary_minus< CPLXSXP, NA > . . . . .	2461
Rcpp::sugar::unary_minus< RTYPE, false > . . . . .	2462
Rcpp::sugar::unary_minus_result_type< RTYPE > . . . . .	2463

Rcpp::sugar::unary_minus_result_type< LGLSXP > . . . . .	2464
Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > . . . . .	2473
Rcpp::internal::UnwindData . . . . .	2477
Vec . . . . .	2489
Rcpp::void_type . . . . .	2577
Rcpp::traits::void_wrap_tag . . . . .	2577
Rcpp::sugar::WhichMax< RTYPE, NA, T > . . . . .	2583
Rcpp::sugar::WhichMax< RTYPE, false, T > . . . . .	2586
Rcpp::sugar::WhichMin< RTYPE, NA, T > . . . . .	2589
Rcpp::sugar::WhichMin< RTYPE, false, T > . . . . .	2592
Rcpp::traits::wrap_type_char_array . . . . .	2597
Rcpp::traits::wrap_type_enum_tag . . . . .	2597
Rcpp::traits::wrap_type_module_object_pointer_tag . . . . .	2598
Rcpp::traits::wrap_type_module_object_tag . . . . .	2598
Rcpp::traits::wrap_type_primitive_tag . . . . .	2598
Rcpp::traits::wrap_type_traits< T > . . . . .	2599
Rcpp::traits::wrap_type_traits< bool > . . . . .	2599
Rcpp::traits::wrap_type_traits< char > . . . . .	2600
Rcpp::traits::wrap_type_traits< char[N]> . . . . .	2601
Rcpp::traits::wrap_type_traits< const char[N]> . . . . .	2602
Rcpp::traits::wrap_type_traits< const int > . . . . .	2602
Rcpp::traits::wrap_type_traits< double > . . . . .	2603
Rcpp::traits::wrap_type_traits< float > . . . . .	2604
Rcpp::traits::wrap_type_traits< int > . . . . .	2604
Rcpp::traits::wrap_type_traits< long > . . . . .	2605
Rcpp::traits::wrap_type_traits< long double > . . . . .	2606
Rcpp::traits::wrap_type_traits< Rbyte > . . . . .	2606
Rcpp::traits::wrap_type_traits< Rcomplex > . . . . .	2607
Rcpp::traits::wrap_type_traits< Rcpp::Date > . . . . .	2608
Rcpp::traits::wrap_type_traits< Rcpp::Datetime > . . . . .	2608
Rcpp::traits::wrap_type_traits< Rcpp::object< T > > . . . . .	2609
Rcpp::traits::wrap_type_traits< Rcpp::String > . . . . .	2610
Rcpp::traits::wrap_type_traits< short > . . . . .	2610
Rcpp::traits::wrap_type_traits< std::complex< double > > . . . . .	2611
Rcpp::traits::wrap_type_traits< std::complex< float > > . . . . .	2612
Rcpp::traits::wrap_type_traits< std::string > . . . . .	2612
Rcpp::traits::wrap_type_traits< std::wstring > . . . . .	2613
Rcpp::traits::wrap_type_traits< unsigned int > . . . . .	2614
Rcpp::traits::wrap_type_traits< unsigned long > . . . . .	2614
Rcpp::traits::wrap_type_traits< unsigned short > . . . . .	2615
Rcpp::traits::wrap_type_traits< wchar_t > . . . . .	2616
Rcpp::traits::wrap_type_unknown_tag . . . . .	2616
Rcpp::traits::zero_type< T > . . . . .	2628



# Chapter 3

## Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Rcpp::traits::__sfinae_types	471
Rcpp::traits::__sfinae_types::__two	473
Rcpp::traits::_has_iterator_helper< T >	474
Rcpp::traits::_has_matrix_interface_helper< T >	476
Rcpp::traits::_has_rtype_helper< T >	478
Rcpp::traits::_is_eigen_helper< T >	480
Rcpp::traits::_is_exporter_helper< T >	482
Rcpp::traits::_is_generator_helper< T >	484
Rcpp::traits::_is_importer_helper< T >	486
Rcpp::traits::_is_sugar_expression_helper< T >	488
Rcpp::traits::_has_iterator_helper< T >::_Wrap_type< U >	491
Rcpp::traits::_has_matrix_interface_helper< T >::_Wrap_type< U >	491
Rcpp::traits::_has_rtype_helper< T >::_Wrap_type< U >	491
Rcpp::traits::_is_eigen_helper< T >::_Wrap_type< U >	492
Rcpp::traits::_is_exporter_helper< T >::_Wrap_type< U >	492
Rcpp::traits::_is_generator_helper< T >::_Wrap_type< U >	492
Rcpp::traits::_is_importer_helper< T >::_Wrap_type< U >	493
Rcpp::traits::_is_sugar_expression_helper< T >::_Wrap_type< U >	493
Rcpp::sugar::All< NA, T >	493
Rcpp::sugar::All< false, T >	497
Rcpp::traits::allowed_matrix_type< bool >	500
Rcpp::traits::allowed_matrix_type< true >	500
Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >	500
Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >	504
Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >	507
Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >	510
Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >	513
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >	517
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >	521
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >	524
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >	528

Rcpp::sugar::Any< NA, T >	531
Rcpp::sugar::Any< false, T >	535
Rcpp::Argument	538
Rcpp::attributes::Argument	539
Rcpp::Armor< T >	543
Rcpp::attributes::Attribute	548
Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy	557
Rcpp::AttributeProxyPolicy< CLASS >	562
Rcpp::stats::BetaGenerator	564
Rcpp::traits::is_convertible< T, U >::Big	566
Rcpp::binary_call< T1, T2, RESULT_TYPE >	567
Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >	570
Rcpp::BindingPolicy< EnvironmentClass >::Binding	573
Rcpp::BindingPolicy< EnvironmentClass >	579
Rcpp::stats::BinomGenerator	580
Rcpp::traits::both< T, U >	583
Cache	584
Rcpp::can_have_na< T >	587
Rcpp::Matrix< RTYPE, StoragePolicy >::can_have_na	588
Rcpp::MatrixBase< RTYPE, na, MATRIX >::can_have_na	589
Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na	590
Rcpp::stats::CauchyGenerator	592
Rcpp::stats::CauchyGenerator_0	594
Rcpp::stats::CauchyGenerator_1	596
Rcpp::sugar::cbind_impl::cbind_sexptype_traits< T >	599
Rcpp::sugar::cbind_impl::cbind_sexptype_traits< SEXP >	600
Rcpp::sugar::cbind_impl::cbind_storage_type< RTYPE >	601
Rcpp::sugar::cbind_impl::cbind_storage_type< LGLSXP >	602
Rcpp::traits::char_type< T >	602
Rcpp::traits::char_type< const char * >	603
Rcpp::traits::char_type< const wchar_t * >	604
Rcpp::ChildVector< T >	605
Rcpp::stats::ChisqGenerator	608
Rcpp::sugar::clamp_operator< RTYPE, NA >	610
Rcpp::sugar::clamp_operator< REALSXP, true >	612
Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >	614
class_< Class >	618
Rcpp::class_Base	638
Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >	647
Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >	651
Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA_RM >	654
Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >	655
Rcpp::sugar::detail::ColMeansReturn< RTYPE >	659
Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >	660
Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA_RM >	663
Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >	664
Rcpp::sugar::detail::ColSumsReturn< RTYPE >	668
Rcpp::attributes::CommentState	669
Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >	671
Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >	675
Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >	679
Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >	683
Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >	688
Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy	694



Rcpp::BindingPolicy< EnvironmentClass >::const_Binding	697
Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy	701
Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy	705
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >	709
Rcpp::MatrixRow< RTYPE >::const_iter_traits	714
Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits	716
Rcpp::ConstMatrixRow< RTYPE >::const_iterator	718
Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy	727
Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy	730
Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy	733
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >	737
Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy	749
Rcpp::ConstInputParameter< T >	752
Rcpp::ConstMatrixColumn< RTYPE >	754
Rcpp::ConstMatrixRow< RTYPE >	760
Rcpp::ConstReferenceInputParameter< T >	766
Rcpp::traits::container_exporter< Container, double >	767
Rcpp::traits::container_exporter< Container, int >	768
Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >	769
Rcpp::traits::ContainerExporter< ContainerTemplate, T >	774
Rcpp::sugar::conversion_to_bool_is_forbidden< x >	776
tinyformat::detail::convertToInt< T, convertible >	778
tinyformat::detail::convertToInt< T, true >	778
Rcpp::sugar::CountInserter< HASH, STORAGE >	779
Rcpp::CppClass	781
Rcpp::attributes::CppExportsGenerator	784
Rcpp::attributes::CppExportsIncludeGenerator	791
Rcpp::CppFinalizer< Class >	797
Rcpp::CppFunction	799
Rcpp::CppFunctionBase	802
Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT_TYPE, Args >	804
Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >	807
Rcpp::CppInheritedMethod< Class, Parent >	809
Rcpp::CppInheritedProperty< Class, Parent >	815
Rcpp::CppMethod< Class >	819
Rcpp::CppObject	822
Rcpp::attributes::CppPackageIncludeGenerator	824
Rcpp::CppProperty< Class >	829
CppProperty_GetConstMethod< Class, PROP >	832
CppProperty_GetConstMethod_SetMethod< Class, PROP >	836
CppProperty_GetConstMethod_SetPointer< Class, PROP >	841
CppProperty_GetMethod< Class, PROP >	846
CppProperty_GetMethod_SetMethod< Class, PROP >	850
CppProperty_GetMethod_SetPointer< Class, PROP >	855
CppProperty_GetPointer_SetMethod< Class, PROP >	860
CppProperty_GetPointer_SetPointer< Class, PROP >	865
CppProperty_GetPointerMethod< Class, PROP >	870
CppProperty_Getter< PROP >	874
CppProperty_Getter_Setter< PROP >	878
Rcpp::algorithm::helpers::ctype< T >	882
Rcpp::algorithm::helpers::CTYPE_CHAR	885
Rcpp::algorithm::helpers::CTYPE_DOUBLE	886
Rcpp::algorithm::helpers::CTYPE_FLOAT	887
Rcpp::algorithm::helpers::ctype_helper< I >	887

Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_CHAR) >	888
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_DOUBLE) >	889
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_FLOAT) >	890
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_INT) >	891
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG) >	892
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG_DOUBLE) >	893
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_SHORT) >	894
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_STRING) >	895
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) >	896
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_INT) >	897
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) >	898
Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) >	899
Rcpp::algorithm::helpers::CTYPE_INT	900
Rcpp::algorithm::helpers::CTYPE_LONG	901
Rcpp::algorithm::helpers::CTYPE_LONG_DOUBLE	901
Rcpp::algorithm::helpers::CTYPE_SHORT	902
Rcpp::algorithm::helpers::CTYPE_STRING	903
Rcpp::algorithm::helpers::CTYPE_UNKNOWN	903
Rcpp::algorithm::helpers::CTYPE_UNSIGNED_CHAR	904
Rcpp::algorithm::helpers::CTYPE_UNSIGNED_INT	905
Rcpp::algorithm::helpers::CTYPE_UNSIGNED_LONG	905
Rcpp::algorithm::helpers::CTYPE_UNSIGNED_SHORT	906
Rcpp::sugar::Cummax< RTYPE, NA, T >	907
Rcpp::sugar::Cummin< RTYPE, NA, T >	910
Rcpp::sugar::Cumprod< RTYPE, NA, T >	913
Rcpp::sugar::Cumsum< RTYPE, NA, T >	916
Rcpp::stats::D0< RTYPE, NA, T >	919
Rcpp::stats::D1< RTYPE, NA, T >	923
Rcpp::stats::D2< RTYPE, NA, T >	927
Rcpp::stats::D3< RTYPE, NA, T >	931
Rcpp::DataFrame_Impl< StoragePolicy >	936
Rcpp::Date	947
Rcpp::Datetime	956
Rcpp::algorithm::helpers::decays_to_ctype< T >	966
Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >	970
Rcpp::sugar::Diag_Maker< RTYPE, NA, T >	973
Rcpp::sugar::diag_result_type_trait< T >	977
Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >	978
Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >	983
Rcpp::sugar::Diff< RTYPE, false, LHS_T >	986
Rcpp::Dimension	990
Rcpp::internal::DimNameProxy	995
Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >	999
Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >	1003
Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >	1007
Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >	1011
Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >	1015
Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >	1019
Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >	1023
Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >	1027
Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >	1031
Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >	1035
Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >	1039
Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >	1042

Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >	1046
Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >	1049
Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >	1053
Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >	1057
Rcpp::DottedPairImpl< CLASS >	1060
Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy	1066
Rcpp::DottedPairProxyPolicy< CLASS >	1072
Rcpp::internal::element_converter< RTYPE >	1073
Rcpp::traits::enable_if< B, T >	1075
Rcpp::traits::enable_if< true, T >	1075
Rcpp::enum_< Enum, Parent >	1076
Rcpp::exception	1080
Rcpp::algorithm::helpers::exp	1085
Rcpp::traits::expands_to_logical< T >	1086
Rcpp::traits::expands_to_logical_impl< RTYPE >	1088
Rcpp::traits::expands_to_logical_impl< LGLSXP >	1090
Rcpp::stats::ExpGenerator	1090
Rcpp::stats::ExpGenerator__rate1	1093
Rcpp::attributes::ExportsGenerator	1095
Rcpp::attributes::ExportsGenerators	1106
Rcpp::traits::Extractor< RTYPE, NA, VECTOR >	1110
tinyformat::detail::is_convertible< T1, T2 >::fail	1110
Rcpp::Fast< VECTOR >	1111
Rcpp::stats::FGenerator_Finite_Finite	1114
Rcpp::stats::FGenerator_Finite_NotFinite	1116
Rcpp::stats::FGenerator_NotFinite_Finite	1119
Rcpp::FieldProxyPolicy< CLASS >::FieldProxy	1121
Rcpp::FieldProxyPolicy< CLASS >	1126
Rcpp::file_exists	1128
Rcpp::file_io_error	1129
Rcpp::file_not_found	1133
Rcpp::attributes::FileInfo	1134
Rcpp::fixed_call< RESULT_TYPE >	1138
Rcpp::sugar::forbidden_conversion< bool >	1140
Rcpp::sugar::forbidden_conversion< true >	1141
tinyformat::detail::FormatArg	1141
tinyformat::FormatList	1145
tinyformat::detail::FormatListN< N >	1147
tinyformat::detail::FormatListN< 0 >	1149
tinyformat::detail::formatValueAsType< T, fmtT, convertible >	1151
tinyformat::detail::formatValueAsType< T, fmtT, true >	1151
Rcpp::attributes::Function	1152
Rcpp::FunctionFinalizer< Class >	1159
Rcpp::stats::GammaGenerator	1161
Rcpp::Generator< T >	1163
Rcpp::internal::generic_element_converter< RTYPE >	1164
Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >	1167
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >	1173
Rcpp::GenericProxy< Proxy >	1183
Rcpp::stats::GeomGenerator	1184
Rcpp::sugar::Grabber< HASH, RTYPE >	1186
Rcpp::traits::has_iterator< T >	1189
Rcpp::traits::has_na< RTYPE >	1190
Rcpp::traits::has_na< CPLXSXP >	1191

Rcpp::traits::has_na< INTSXP >	1192
Rcpp::traits::has_na< LGLSXP >	1193
Rcpp::traits::has_na< REALSXP >	1194
Rcpp::traits::has_na< STRSXP >	1195
Rcpp::sugar::cbind_impl::detail::has_stored_type< T >	1196
Rcpp::sugar::Head< RTYPE, NA, T >	1198
Rcpp::stats::HyperGenerator	1201
Rcpp::traits::identity< T >	1203
Rcpp::traits::if_< COND, LHS, RHS >	1204
Rcpp::traits::if_< false, LHS, RHS >	1205
Rcpp::sugar::ifElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >	1206
Rcpp::sugar::ifElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >	1211
Rcpp::sugar::ifElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >	1215
Rcpp::sugar::ifElse_Primitive_Primitive< RTYPE, false, COND_T >	1219
Rcpp::sugar::ifElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >	1223
Rcpp::sugar::ifElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >	1227
Rcpp::sugar::ifElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >	1231
Rcpp::sugar::ifElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >	1236
Rcpp::sugar::In< RTYPE, TABLE_T >	1240
Rcpp::sugar::IndexHash< RTYPE >	1242
Rcpp::traits::init_type< RTYPE >	1253
Rcpp::traits::init_type< LGLSXP >	1253
Rcpp::traits::init_type< STRSXP >	1254
Rcpp::traits::input_parameter< T >	1255
Rcpp::traits::input_parameter< const T & >	1255
Rcpp::traits::input_parameter< const T >	1256
Rcpp::traits::input_parameter< T & >	1257
Rcpp::InputParameter< T >	1258
Rcpp::sugar::InSet< HASH >	1259
Rcpp::traits::int2type< I >	1261
Rcpp::traits::integral_constant< _T, _V >	1262
interrupt_exception	1264
Rcpp::internal::InterruptedException	1266
Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >	1266
Rcpp::traits::is_arithmetic< typename >	1270
Rcpp::traits::is_arithmetic< const double >	1271
Rcpp::traits::is_arithmetic< const float >	1272
Rcpp::traits::is_arithmetic< const int >	1273
Rcpp::traits::is_arithmetic< const long >	1274
Rcpp::traits::is_arithmetic< const long double >	1275
Rcpp::traits::is_arithmetic< const short >	1276
Rcpp::traits::is_arithmetic< const unsigned int >	1277
Rcpp::traits::is_arithmetic< const unsigned long >	1278
Rcpp::traits::is_arithmetic< const unsigned short >	1279
Rcpp::traits::is_arithmetic< double >	1280
Rcpp::traits::is_arithmetic< float >	1281
Rcpp::traits::is_arithmetic< int >	1282
Rcpp::traits::is_arithmetic< long >	1283
Rcpp::traits::is_arithmetic< long double >	1284
Rcpp::traits::is_arithmetic< short >	1285
Rcpp::traits::is_arithmetic< unsigned int >	1286
Rcpp::traits::is_arithmetic< unsigned long >	1287
Rcpp::traits::is_arithmetic< unsigned short >	1288
Rcpp::traits::is_bool< typename >	1289

Rcpp::traits::is_bool< bool > . . . . .	1290
Rcpp::traits::is_bool< const bool > . . . . .	1291
Rcpp::traits::is_bool< volatile bool > . . . . .	1292
Rcpp::traits::is_const< typename > . . . . .	
Type properties [4.5.3] . . . . .	1293
Rcpp::traits::is_const< _Tp const > . . . . .	1294
Rcpp::traits::is_convertible< T, U > . . . . .	1295
tinyformat::detail::is_convertible< T1, T2 > . . . . .	1297
Rcpp::traits::is_eigen_base< T > . . . . .	1299
Rcpp::traits::is_exporter< T > . . . . .	1300
Rcpp::traits::is_generator< T > . . . . .	1302
Rcpp::traits::is_importer< T > . . . . .	1303
Rcpp::traits::is_module_object< T > . . . . .	1304
Rcpp::traits::is_named< T > . . . . .	1305
Rcpp::traits::is_named< named_object< T > > . . . . .	1306
Rcpp::traits::is_named< Rcpp::Argument > . . . . .	1307
Rcpp::traits::is_pointer< T > . . . . .	1308
Rcpp::traits::is_pointer< T * > . . . . .	1309
Rcpp::traits::is_primitive< T > . . . . .	1310
Rcpp::traits::is_reference< typename > . . . . .	1311
Rcpp::traits::is_reference< _Tp & > . . . . .	1313
Rcpp::traits::is_sugar_expression< T > . . . . .	1314
Rcpp::sugar::is_sugar_vector< T > . . . . .	1315
Rcpp::sugar::is_sugar_vector< Rcpp::Vector< RTYPE > > . . . . .	1316
Rcpp::traits::is_trivial< RTYPE > . . . . .	1317
Rcpp::traits::is_trivial< EXPRXP > . . . . .	1318
Rcpp::traits::is_trivial< VECSXP > . . . . .	1319
tinyformat::detail::is_wchar< T > . . . . .	1320
tinyformat::detail::is_wchar< const wchar_t * > . . . . .	1321
tinyformat::detail::is_wchar< const wchar_t[n]> . . . . .	1321
tinyformat::detail::is_wchar< wchar_t * > . . . . .	1322
tinyformat::detail::is_wchar< wchar_t[n]> . . . . .	1322
Rcpp::traits::is_wide_string< T > . . . . .	1322
Rcpp::traits::is_wide_string< char > . . . . .	1324
Rcpp::traits::is_wide_string< const char * > . . . . .	1325
Rcpp::traits::is_wide_string< const wchar_t * > . . . . .	1326
Rcpp::traits::is_wide_string< wchar_t > . . . . .	1327
Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE > . . . . .	1328
Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE > . . . . .	1331
Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE > . . . . .	1334
Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE > . . . . .	1337
Rcpp::sugar::IsNa_Vector_is_na< T > . . . . .	1340
Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE > . . . . .	1343
Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS > . . . . .	1346
Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS > . . . . .	1355
Rcpp::MatrixRow< RTYPE >::iter_traits . . . . .	1363
Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits . . . . .	1365
Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator . . . . .	1367
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 > . . . . .	1379
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > > . . . . .	1384
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 > . . . . .	1388
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > > . . . . .	1393
Rcpp::sugar::Lapply< RTYPE, NA, T, Function > . . . . .	1397

Rcpp::sugar::Lazy< T, EXPR >	1401
Rcpp::internal::LazyVector< VECTOR >	1402
Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >	1405
Rcpp::ListOf< T >	1408
Rcpp::stats::LNormGenerator	1417
Rcpp::stats::LNormGenerator_0	1419
Rcpp::stats::LNormGenerator_1	1421
Rcpp::algorithm::helpers::log	1423
Rcpp::stats::LogisGenerator	1425
Rcpp::stats::LogisGenerator_0	1427
Rcpp::stats::LogisGenerator_1	1429
Rcpp::LongjumpException	1431
Rcpp::sugar::LowerTri< RTYPE, NA, T >	1433
Rcpp::lsinfo	1439
Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, is_container >	1439
Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true >	1440
Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true >	1442
Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, false >	1443
Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, true >	1445
Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >	1447
Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >	1450
Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >	1453
Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >	1457
Rcpp::Matrix< RTYPE, StoragePolicy >	1462
Rcpp::traits::matrix_interface< T >	1486
Rcpp::sugar::cbind_impl::detail::matrix_return< T, is_container >	1487
Rcpp::sugar::cbind_impl::matrix_return< T, B >	1487
Rcpp::sugar::cbind_impl::detail::matrix_return< bool, false >	1488
Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Matrix< LGLSXP >, true >	1489
Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Vector< LGLSXP >, true >	1489
Rcpp::sugar::cbind_impl::detail::matrix_return< T, false >	1490
Rcpp::sugar::cbind_impl::matrix_return< T, false >	1491
Rcpp::sugar::cbind_impl::detail::matrix_return< T, true >	1493
Rcpp::MatrixBase< RTYPE, na, MATRIX >	1494
Rcpp::MatrixColumn< RTYPE >	1499
Rcpp::MatrixRow< RTYPE >	1507
Rcpp::sugar::Max< RTYPE, NA, T >	1516
Rcpp::sugar::Max< RTYPE, false, T >	1518
Rcpp::sugar::Mean< RTYPE, NA, T >	1520
Rcpp::sugar::Mean< CPLXSXP, NA, T >	1524
Rcpp::sugar::Mean< INTSXP, NA, T >	1527
Rcpp::sugar::Mean< LGLSXP, NA, T >	1530
Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >	1532
Rcpp::sugar::Median< RTYPE, false, T, NA_RM >	1536
Rcpp::sugar::Median< RTYPE, NA, T, true >	1538
Rcpp::sugar::Median< STRSXP, false, T, true >	1541
Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >	1544
Rcpp::sugar::Median< STRSXP, NA, T, true >	1547
Rcpp::sugar::Min< RTYPE, NA, T >	1549
Rcpp::sugar::Min< RTYPE, false, T >	1551
Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >	1553
Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >	1557
Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >	1561
Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >	1565

Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >	1569
Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >	1573
Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >	1577
Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >	1581
Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >	1585
Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >	1589
Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >	1593
Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >	1596
Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >	1600
Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >	1603
Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >	1607
Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >	1611
Rcpp::Module	1614
Rcpp::traits::module_wrap_traits< T >	1624
Rcpp::traits::module_wrap_traits< T * >	1625
Rcpp::traits::module_wrap_traits< void >	1626
Rcpp::Na_Proxy	1626
Rcpp::traits::named_object< T >	1631
Rcpp::traits::named_object< SEXP >	1632
Rcpp::internal::NamedPlaceholder	1635
Rcpp::NamesProxyPolicy< CLASS >::NamesProxy	1636
Rcpp::NamesProxyPolicy< CLASS >	1641
Rcpp::stats::NBinomGenerator	1642
Rcpp::stats::NBinomGenerator_Mu	1645
Rcpp::stats::NChisqGenerator	1647
Rcpp::traits::needs_protection< T >	1650
Rcpp::traits::needs_protection< SEXP >	1651
Rcpp::sugar::negate< NA >	1652
Rcpp::sugar::negate< false >	1653
Rcpp::sugar::Negate_SingleLogicalResult< NA, T >	1653
Rcpp::newDatetimeVector	1657
Rcpp::newDateVector	1661
Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::no	1666
Rcpp::no_init_matrix	1666
Rcpp::no_init_vector	1668
Rcpp::no_such_env	1670
Rcpp::sugar::Nona< RTYPE, NA, VECTOR >	1672
Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >	1676
Rcpp::sugar::NonaPrimitive< T >	1680
Rcpp::NoProtectStorage< CLASS >	1681
Rcpp::traits::normal_wrap_tag	1684
Rcpp::stats::NormGenerator	1685
Rcpp::stats::NormGenerator__mean0	1687
Rcpp::stats::NormGenerator__mean0__sd1	1690
Rcpp::stats::NormGenerator__sd1	1692
Rcpp::sugar::not_< RTYPE, NA >	1694
Rcpp::sugar::not_< CPLXSXP, false >	1695
Rcpp::sugar::not_< CPLXSXP, NA >	1696
Rcpp::sugar::not_< REALSXP, false >	1697
Rcpp::sugar::not_< REALSXP, NA >	1698
Rcpp::sugar::not_< RTYPE, false >	1699
Rcpp::sugar::Not_Vector< RTYPE, NA, T >	1700
Rcpp::Nullable< T >	1705
Rcpp::traits::num2type< N >	1712



Rcpp::object< T >	1712
Rcpp::oldDatetimeVector	1715
Rcpp::oldDateVector	1717
Rcpp::traits::one_type< T >	1718
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >	1720
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >	1724
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >	1727
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >	1730
Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >	1733
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >	1737
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >	1741
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >	1744
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >	1748
Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >	1751
Rcpp::stats::P0< RTYPE, NA, T >	1758
Rcpp::stats::P1< RTYPE, NA, T >	1762
Rcpp::stats::P2< RTYPE, NA, T >	1766
Rcpp::stats::P3< RTYPE, NA, T >	1771
Rcpp::attributes::Param	1775
Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >	1779
Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >	1783
Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >	1787
Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >	1791
Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >	1795
Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >	1799
Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >	1802
Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >	1806
Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >	1810
Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >	1814
Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >	1817
Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >	1821
Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >	1824
Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >	1828
Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >	1832
Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >	1836
Rcpp::sugar::pmax_op< RTYPE, LHS_NA, RHS_NA >	1839
Rcpp::sugar::pmax_op< INTSXP, LHS_NA, RHS_NA >	1840
Rcpp::sugar::pmax_op< REALSXP, false, false >	1840
Rcpp::sugar::pmax_op< REALSXP, false, true >	1841
Rcpp::sugar::pmax_op< REALSXP, true, false >	1842
Rcpp::sugar::pmax_op< REALSXP, true, true >	1843
Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >	1844
Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >	1845
Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >	1847
Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >	1851
Rcpp::sugar::pmin_op< RTYPE, LHS_NA, RHS_NA >	1854
Rcpp::sugar::pmin_op< INTSXP, LHS_NA, RHS_NA >	1854
Rcpp::sugar::pmin_op< REALSXP, false, false >	1855
Rcpp::sugar::pmin_op< REALSXP, false, true >	1856
Rcpp::sugar::pmin_op< REALSXP, true, false >	1857
Rcpp::sugar::pmin_op< REALSXP, true, true >	1857
Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >	1858
Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >	1860
Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >	1862



Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >	1865
Rcpp::traits::pointer_wrap_tag	1868
Rcpp::stats::PoissonGenerator	1869
Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >	1871
Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >	1875
Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >	1878
Rcpp::PreserveStorage< CLASS >	1881
Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy	1885
Rcpp::ProtectedProxyPolicy< XPtrClass >	1889
Rcpp::traits::proxy_based_const_iterator< RTYPE, StoragePolicy >	1890
Rcpp::traits::proxy_based_iterator< RTYPE, StoragePolicy >	1891
Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >	1892
Rcpp::stats::Q0< RTYPE, NA, T >	1897
Rcpp::stats::Q1< RTYPE, NA, T >	1901
Rcpp::stats::Q2< RTYPE, NA, T >	1905
Rcpp::stats::Q3< RTYPE, NA, T >	1910
Rcpp::traits::expands_to_logical_impl< LGLSXP >::r_expands_to_logical	1914
Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_matrix_interface	1915
Rcpp::traits::r_sexptype_needscastr< T >	1915
Rcpp::traits::r_sexptype_needscastr< double >	1916
Rcpp::traits::r_sexptype_needscastr< int >	1917
Rcpp::traits::r_sexptype_needscastr< Rbyte >	1918
Rcpp::traits::r_sexptype_needscastr< Rcomplex >	1919
Rcpp::traits::r_sexptype_traits< T >	1920
Rcpp::traits::r_sexptype_traits< bool >	1921
Rcpp::traits::r_sexptype_traits< const double >	1922
Rcpp::traits::r_sexptype_traits< const int >	1923
Rcpp::traits::r_sexptype_traits< double >	1924
Rcpp::traits::r_sexptype_traits< float >	1924
Rcpp::traits::r_sexptype_traits< int >	1925
Rcpp::traits::r_sexptype_traits< long >	1926
Rcpp::traits::r_sexptype_traits< long double >	1927
Rcpp::traits::r_sexptype_traits< Rbyte >	1927
Rcpp::traits::r_sexptype_traits< Rcomplex >	1928
Rcpp::traits::r_sexptype_traits< Rcpp::Date >	1929
Rcpp::traits::r_sexptype_traits< Rcpp::Datetime >	1930
Rcpp::traits::r_sexptype_traits< Rcpp::String >	1930
Rcpp::traits::r_sexptype_traits< short >	1931
Rcpp::traits::r_sexptype_traits< std::complex< double > >	1932
Rcpp::traits::r_sexptype_traits< std::complex< float > >	1933
Rcpp::traits::r_sexptype_traits< std::string >	1933
Rcpp::traits::r_sexptype_traits< unsigned int >	1934
Rcpp::traits::r_sexptype_traits< unsigned long >	1935
Rcpp::traits::r_sexptype_traits< unsigned short >	1936
Rcpp::Matrix< RTYPE, StoragePolicy >::r_type	1936
Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_type	1938
Rcpp::VectorBase< RTYPE, na, VECTOR >::r_type	1939
Rcpp::traits::r_type_enum_tag	1940
Rcpp::traits::r_type_generic_tag	1940
Rcpp::traits::r_type_module_object_const_pointer_tag	1940
Rcpp::traits::r_type_module_object_const_reference_tag	1941
Rcpp::traits::r_type_module_object_pointer_tag	1941
Rcpp::traits::r_type_module_object_reference_tag	1941
Rcpp::traits::r_type_module_object_tag	1942

<a href="#">Rcpp::traits::r_type_pair_tag</a>	1942
<a href="#">Rcpp::traits::r_type_pairstring_generic_tag</a>	1942
<a href="#">Rcpp::traits::r_type_pairstring_primitive_tag</a>	1943
<a href="#">Rcpp::traits::r_type_pairstring_string_tag</a>	1943
<a href="#">Rcpp::traits::r_type_primitive_tag</a>	1943
<a href="#">Rcpp::traits::r_type_RcppString_tag</a>	1944
<a href="#">Rcpp::traits::r_type_string_tag</a>	1944
<a href="#">Rcpp::traits::r_type_traits&lt; T &gt;</a>	1944
<a href="#">Rcpp::traits::r_type_traits&lt; bool &gt;</a>	1945
<a href="#">Rcpp::traits::r_type_traits&lt; char &gt;</a>	1946
<a href="#">Rcpp::traits::r_type_traits&lt; const char * &gt;</a>	1946
<a href="#">Rcpp::traits::r_type_traits&lt; const double &gt;</a>	1947
<a href="#">Rcpp::traits::r_type_traits&lt; const int &gt;</a>	1948
<a href="#">Rcpp::traits::r_type_traits&lt; const wchar_t * &gt;</a>	1948
<a href="#">Rcpp::traits::r_type_traits&lt; double &gt;</a>	1949
<a href="#">Rcpp::traits::r_type_traits&lt; float &gt;</a>	1950
<a href="#">Rcpp::traits::r_type_traits&lt; int &gt;</a>	1950
<a href="#">Rcpp::traits::r_type_traits&lt; long &gt;</a>	1951
<a href="#">Rcpp::traits::r_type_traits&lt; long double &gt;</a>	1952
<a href="#">Rcpp::traits::r_type_traits&lt; Rbyte &gt;</a>	1952
<a href="#">Rcpp::traits::r_type_traits&lt; Rcomplex &gt;</a>	1953
<a href="#">Rcpp::traits::r_type_traits&lt; Rcpp::Date &gt;</a>	1954
<a href="#">Rcpp::traits::r_type_traits&lt; Rcpp::Datetime &gt;</a>	1954
<a href="#">Rcpp::traits::r_type_traits&lt; Rcpp::object&lt; T &gt; &gt;</a>	1955
<a href="#">Rcpp::traits::r_type_traits&lt; Rcpp::String &gt;</a>	1956
<a href="#">Rcpp::traits::r_type_traits&lt; short &gt;</a>	1956
<a href="#">Rcpp::traits::r_type_traits&lt; std::complex&lt; double &gt; &gt;</a>	1957
<a href="#">Rcpp::traits::r_type_traits&lt; std::complex&lt; float &gt; &gt;</a>	1958
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const KEY, VALUE &gt; &gt;</a>	1958
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, bool &gt; &gt;</a>	1959
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, char &gt; &gt;</a>	1960
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, const int &gt; &gt;</a>	1961
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, double &gt; &gt;</a>	1961
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, float &gt; &gt;</a>	1962
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, int &gt; &gt;</a>	1963
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, long &gt; &gt;</a>	1963
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, long double &gt; &gt;</a>	1964
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, Rbyte &gt; &gt;</a>	1965
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, Rcomplex &gt; &gt;</a>	1965
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, Rcpp::Date &gt; &gt;</a>	1966
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, Rcpp::Datetime &gt; &gt;</a>	1967
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, short &gt; &gt;</a>	1967
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, std::complex&lt; double &gt; &gt; &gt;</a>	1968
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, std::complex&lt; float &gt; &gt; &gt;</a>	1969
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, std::string &gt; &gt;</a>	1969
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, std::wstring &gt; &gt;</a>	1970
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, T &gt; &gt;</a>	1971
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, unsigned int &gt; &gt;</a>	1971
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, unsigned long &gt; &gt;</a>	1972
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, unsigned short &gt; &gt;</a>	1973
<a href="#">Rcpp::traits::r_type_traits&lt; std::pair&lt; const std::string, wchar_t &gt; &gt;</a>	1974
<a href="#">Rcpp::traits::r_type_traits&lt; std::string &gt;</a>	1974
<a href="#">Rcpp::traits::r_type_traits&lt; std::wstring &gt;</a>	1975
<a href="#">Rcpp::traits::r_type_traits&lt; unsigned int &gt;</a>	1976

Rcpp::traits::r_type_traits< unsigned long >	1976
Rcpp::traits::r_type_traits< unsigned short >	1977
Rcpp::traits::r_type_traits< wchar_t >	1978
Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >	1978
Rcpp::traits::r_vector_cache_type< RTYPE, StoragePolicy >	1983
Rcpp::traits::r_vector_cache_type< EXPRSXP, StoragePolicy >	1984
Rcpp::traits::r_vector_cache_type< STRSXP, StoragePolicy >	1984
Rcpp::traits::r_vector_cache_type< VECSXP, StoragePolicy >	1985
Rcpp::traits::r_vector_const_iterator< RTYPE, StoragePolicy >	1986
Rcpp::traits::r_vector_const_iterator< EXPRSXP, StoragePolicy >	1987
Rcpp::traits::r_vector_const_iterator< STRSXP, StoragePolicy >	1988
Rcpp::traits::r_vector_const_iterator< VECSXP, StoragePolicy >	1990
Rcpp::traits::r_vector_const_proxy< RTYPE, StoragePolicy >	1991
Rcpp::traits::r_vector_const_proxy< EXPRSXP, StoragePolicy >	1992
Rcpp::traits::r_vector_const_proxy< STRSXP, StoragePolicy >	1992
Rcpp::traits::r_vector_const_proxy< VECSXP, StoragePolicy >	1993
Rcpp::traits::r_vector_element_converter< RTYPE >	1994
Rcpp::traits::r_vector_element_converter< EXPRSXP >	1995
Rcpp::traits::r_vector_element_converter< STRSXP >	1996
Rcpp::traits::r_vector_element_converter< VECSXP >	1996
Rcpp::traits::r_vector_iterator< RTYPE, StoragePolicy >	1997
Rcpp::traits::r_vector_iterator< EXPRSXP, StoragePolicy >	1998
Rcpp::traits::r_vector_iterator< STRSXP, StoragePolicy >	1999
Rcpp::traits::r_vector_iterator< VECSXP, StoragePolicy >	2000
Rcpp::traits::r_vector_name_proxy< RTYPE, StoragePolicy >	2001
Rcpp::traits::r_vector_name_proxy< EXPRSXP, StoragePolicy >	2002
Rcpp::traits::r_vector_name_proxy< STRSXP, StoragePolicy >	2003
Rcpp::traits::r_vector_name_proxy< VECSXP, StoragePolicy >	2004
Rcpp::traits::r_vector_proxy< RTYPE, StoragePolicy >	2005
Rcpp::traits::r_vector_proxy< EXPRSXP, StoragePolicy >	2005
Rcpp::traits::r_vector_proxy< STRSXP, StoragePolicy >	2006
Rcpp::traits::r_vector_proxy< VECSXP, StoragePolicy >	2007
Rcpp::Range	2008
Rcpp::sugar::Range< RTYPE, NA, T >	2014
Rcpp::sugar::Range< RTYPE, false, T >	2016
Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >	2019
Rcpp::VectorBase< RTYPE, na, VECTOR >::rcpp_sugar_expression	2024
Rcpp::ReferenceInputParameter< T >	2024
Rcpp::traits::remove_const< _Tp >	
Const-volatile modifications [4.7.1]	2026
Rcpp::traits::remove_const< _Tp const >	2027
Rcpp::traits::remove_const_and_reference< T >	2028
Rcpp::traits::remove_reference< _Tp >	
Reference modifications [4.7.2]	2028
Rcpp::traits::remove_reference< _Tp & >	2029
Rcpp::sugar::RemoveFromSet< SET >	2030
Rcpp::sugar::Rep< RTYPE, NA, T >	2032
Rcpp::sugar::Rep_each< RTYPE, NA, T >	2035
Rcpp::sugar::Rep_len< RTYPE, NA, T >	2039
Rcpp::sugar::Rep_Single< T >	2042
Rcpp::result< T >	2045
Rcpp::sugar::median_detail::result< RTYPE >	2046
Rcpp::sugar::median_detail::result< INTSXP >	2047
Rcpp::sugar::median_detail::result< STRSXP >	2048

Rcpp::traits::result_of< T >	2049
Rcpp::traits::result_of< RESULT_TYPE(*) (INPUT_TYPE) >	2050
Rcpp::traits::result_of< RESULT_TYPE(*) (U1, U2) >	2051
Rcpp::traits::result_of< RESULT_TYPE(*) (U1, U2, U3) >	2051
Rcpp::sugar::Rev< RTYPE, NA, T >	2052
Rcpp::attributes::RExportsGenerator	2055
Rcpp::RNGScope	2059
Rcpp::RObjectMethods< Class >	2061
Rcpp::Rostream< OUTPUT >	2063
Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >	2065
Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >	2069
Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA_RM >	2072
Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >	2073
Rcpp::sugar::detail::RowMeansReturn< RTYPE >	2077
Rcpp::sugar::detail::RowMeansReturn< CPLXSXP >	2078
Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >	2079
Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA_RM >	2083
Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >	2084
Rcpp::sugar::detail::RowSumsReturn< RTYPE >	2087
Rcpp::sugar::detail::RowSumsReturn< LGLSXP >	2089
Rcpp::Rstreambuf< OUTPUT >	2090
Rcpp::algorithm::helpers::rtype< T >	2093
Rcpp::algorithm::helpers::rtype_helper< T >	2095
Rcpp::algorithm::helpers::rtype_helper< double >	2096
Rcpp::algorithm::helpers::rtype_helper< int >	2097
Rcpp::rule	2099
Rcpp::S4_CppConstructor< Class >	2101
Rcpp::S4_CppOverloadedMethods< Class >	2103
Rcpp::S4_field< Class >	2107
Rcpp::traits::same_type< T, U >	2109
Rcpp::traits::same_type< T, T >	2111
Rcpp::sugar::Supply< RTYPE, NA, T, Function, NO_CONVERSION >	2112
Rcpp::sugar::Supply< RTYPE, NA, T, Function, true >	2116
Rcpp::sugar::supply_application_result_of< Function, SugarExpression >	2120
Rcpp::sugar::cbind_impl::scalar< RTYPE >	2121
Rcpp::sugar::cbind_impl::ScalarBindable< T >	2122
Rcpp::sugar::Sd< RTYPE, NA, T >	2125
Rcpp::sugar::SelfHash< RTYPE >	2128
Rcpp::sugar::SelfInserter< HASH, STORAGE >	2134
Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >	2136
Rcpp::sugar::SeqLen	2140
Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >	2142
Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >	2145
Rcpp::Shelter< T >	2148
Rcpp::Shield< T >	2151
Rcpp::sugar::Sign< RTYPE, NA, T >	2154
Rcpp::sugar::sign_impl< NA, RTYPE >	2158
Rcpp::sugar::sign_impl< false, RTYPE >	2159
Rcpp::SignedConstructor< Class >	2160
Rcpp::SignedFactory< Class >	2163
Rcpp::SignedMethod< Class >	2165
Rcpp::stats::SignRankGenerator	2170
Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >	2172
Rcpp::SingleLogicalResult< NA, T >	2178

Rcpp::sugar::SingleLogicalResult< NA, T >	2179
Rcpp::SlotProxyPolicy< CLASS >::SlotProxy	2188
Rcpp::SlotProxyPolicy< CLASS >	2193
Rcpp::attributes::SourceFileAttributes	2194
Rcpp::attributes::SourceFileAttributesParser	2198
Rcpp::algorithm::helpers::sqrt	2207
Rcpp::state	2209
Rcpp::traits::storage_type< RTYPE >	2212
Rcpp::traits::storage_type< CPLXSXP >	2213
Rcpp::traits::storage_type< INTSXP >	2213
Rcpp::traits::storage_type< LGLSXP >	2214
Rcpp::traits::storage_type< RAWSXP >	2215
Rcpp::traits::storage_type< REALSXP >	2215
Rcpp::String	2216
Rcpp::internal::string_element_converter< RTYPE >	2259
Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >	2261
Rcpp::internal::string_proxy< RTYPE, StoragePolicy >	2270
Rcpp::StringTransformer< UnaryOperator >	2291
Rcpp::SubMatrix< RTYPE >	2293
Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >	2299
tinyformat::detail::is_convertible< T1, T2 >::succeed	2311
Rcpp::sugar::sugar_const_iterator_type< T >	2311
Rcpp::sugar::sugar_const_iterator_type< CharacterVector >	2312
Rcpp::sugar::sugar_const_iterator_type< Rcpp::Vector< RTYPE > >	2313
Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >	2314
Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >	2317
Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >	2320
Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >	2323
Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >	2326
Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >	2330
Rcpp::sugar::SugarIterator< T >	2333
Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >	2341
Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >	2344
Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >	2347
Rcpp::sugar::Sum< RTYPE, NA, T >	2351
Rcpp::sugar::Sum< REALSXP, NA, T >	2354
Rcpp::sugar::Sum< RTYPE, false, T >	2356
Rcpp::SuspendRNGSynchronizationScope	2359
Rcpp::sugar::Table< RTYPE, TABLE_T >	2360
Rcpp::TagProxyPolicy< XPtrClass >::TagProxy	2364
Rcpp::TagProxyPolicy< XPtrClass >	2367
Rcpp::sugar::Tail< RTYPE, NA, T >	2369
Rcpp::stats::TGenerator	2372
Rcpp::Timer	2374
Timer	2378
Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >	2384
Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >	2388
Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >	2392
Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >	2396
Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >	2400
Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >	2404
Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >	2407
Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >	2411
Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >	2415

Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >	2419
Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >	2422
Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >	2426
Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >	2429
Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >	2433
Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >	2437
Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >	2441
Rcpp::tinfo	2444
Rcpp::attributes::Type	2446
Rcpp::thead	2451
Rcpp::traits::un_pointer< T >	2453
Rcpp::traits::un_pointer< object< T > >	2454
Rcpp::traits::un_pointer< T * >	2455
Rcpp::unary_call< T, RESULT_TYPE >	2456
Rcpp::sugar::unary_minus< RTYPE, NA >	2459
Rcpp::sugar::unary_minus< CPLXSXP, false >	2460
Rcpp::sugar::unary_minus< CPLXSXP, NA >	2461
Rcpp::sugar::unary_minus< RTYPE, false >	2462
Rcpp::sugar::unary_minus_result_type< RTYPE >	2463
Rcpp::sugar::unary_minus_result_type< LGLSXP >	2464
Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >	2465
Rcpp::stats::UnifGenerator	2469
Rcpp::stats::UnifGenerator__0__1	2472
Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >	2473
Rcpp::internal::UnwindData	2477
Rcpp::sugar::UpperTri< RTYPE, NA, T >	2477
Rcpp::sugar::Var< RTYPE, NA, T >	2483
Rcpp::sugar::Var< CPLXSXP, NA, T >	2486
Vec	2489
Rcpp::Vector< RTYPE, StoragePolicy >	2494
Rcpp::VectorBase< RTYPE, na, VECTOR >	2561
Rcpp::sugar::Vectorized< Func, NA, VEC >	2567
Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >	2571
Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >	2574
Rcpp::void_type	2577
Rcpp::traits::void_wrap_tag	2577
Rcpp::stats::WeibullGenerator	2578
Rcpp::stats::WeibullGenerator__scale1	2580
Rcpp::sugar::WhichMax< RTYPE, NA, T >	2583
Rcpp::sugar::WhichMax< RTYPE, false, T >	2586
Rcpp::sugar::WhichMin< RTYPE, NA, T >	2589
Rcpp::sugar::WhichMin< RTYPE, false, T >	2592
Rcpp::stats::WilcoxGenerator	2595
Rcpp::traits::wrap_type_char_array	2597
Rcpp::traits::wrap_type_enum_tag	2597
Rcpp::traits::wrap_type_module_object_pointer_tag	2598
Rcpp::traits::wrap_type_module_object_tag	2598
Rcpp::traits::wrap_type_primitive_tag	2598
Rcpp::traits::wrap_type_traits< T >	2599
Rcpp::traits::wrap_type_traits< bool >	2599
Rcpp::traits::wrap_type_traits< char >	2600
Rcpp::traits::wrap_type_traits< char[N]>	2601
Rcpp::traits::wrap_type_traits< const char[N]>	2602
Rcpp::traits::wrap_type_traits< const int >	2602

Rcpp::traits::wrap_type_traits< double > . . . . .	2603
Rcpp::traits::wrap_type_traits< float > . . . . .	2604
Rcpp::traits::wrap_type_traits< int > . . . . .	2604
Rcpp::traits::wrap_type_traits< long > . . . . .	2605
Rcpp::traits::wrap_type_traits< long double > . . . . .	2606
Rcpp::traits::wrap_type_traits< Rbyte > . . . . .	2606
Rcpp::traits::wrap_type_traits< Rcomplex > . . . . .	2607
Rcpp::traits::wrap_type_traits< Rcpp::Date > . . . . .	2608
Rcpp::traits::wrap_type_traits< Rcpp::Datetime > . . . . .	2608
Rcpp::traits::wrap_type_traits< Rcpp::object< T > > . . . . .	2609
Rcpp::traits::wrap_type_traits< Rcpp::String > . . . . .	2610
Rcpp::traits::wrap_type_traits< short > . . . . .	2610
Rcpp::traits::wrap_type_traits< std::complex< double > > . . . . .	2611
Rcpp::traits::wrap_type_traits< std::complex< float > > . . . . .	2612
Rcpp::traits::wrap_type_traits< std::string > . . . . .	2612
Rcpp::traits::wrap_type_traits< std::wstring > . . . . .	2613
Rcpp::traits::wrap_type_traits< unsigned int > . . . . .	2614
Rcpp::traits::wrap_type_traits< unsigned long > . . . . .	2614
Rcpp::traits::wrap_type_traits< unsigned short > . . . . .	2615
Rcpp::traits::wrap_type_traits< wchar_t > . . . . .	2616
Rcpp::traits::wrap_type_unknown_tag . . . . .	2616
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > . . . . .	2617
Rcpp::traits::zero_type< T > . . . . .	2628





# Chapter 4

## File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">inst/examples/Attributes/cppFunction.R</a>	2631
<a href="#">inst/examples/Attributes/Depends.cpp</a>	2631
<a href="#">inst/examples/Attributes/Export.cpp</a>	2632
<a href="#">inst/examples/Attributes/sourceCpp.R</a>	2634
<a href="#">inst/examples/ConvolveBenchmarks/convolve10_cpp.cpp</a>	2634
<a href="#">inst/examples/ConvolveBenchmarks/convolve10_cpp.h</a>	2635
<a href="#">inst/examples/ConvolveBenchmarks/convolve11_cpp.cpp</a>	2636
<a href="#">inst/examples/ConvolveBenchmarks/convolve12_cpp.cpp</a>	2637
<a href="#">inst/examples/ConvolveBenchmarks/convolve13_cpp.cpp</a>	2638
<a href="#">inst/examples/ConvolveBenchmarks/convolve14_cpp.cpp</a>	2639
<a href="#">inst/examples/ConvolveBenchmarks/convolve2_c.c</a>	2640
<a href="#">inst/examples/ConvolveBenchmarks/convolve3_cpp.cpp</a>	2641
<a href="#">inst/examples/ConvolveBenchmarks/convolve4_cpp.cpp</a>	2642
<a href="#">inst/examples/ConvolveBenchmarks/convolve5_cpp.cpp</a>	2643
<a href="#">inst/examples/ConvolveBenchmarks/convolve7_c.c</a>	2644
<a href="#">inst/examples/ConvolveBenchmarks/convolve8_cpp.cpp</a>	2645
<a href="#">inst/examples/ConvolveBenchmarks/convolve9_cpp.cpp</a>	2646
<a href="#">inst/examples/ConvolveBenchmarks/exampleRCode.r</a>	2647
<a href="#">inst/examples/ConvolveBenchmarks/loopmacro.h</a>	2647
<a href="#">inst/examples/ConvolveBenchmarks/overhead.r</a>	2648
<a href="#">inst/examples/ConvolveBenchmarks/overhead_1.cpp</a>	2648
<a href="#">inst/examples/ConvolveBenchmarks/overhead_2.c</a>	2650
<a href="#">inst/examples/FastLM/benchmark.r</a>	2651
<a href="#">inst/examples/FastLM/benchmarkLongley.r</a>	2651
<a href="#">inst/examples/FastLM/fastLMviaArmadillo.r</a>	2651
<a href="#">inst/examples/FastLM/fastLMviaGSL.r</a>	2651
<a href="#">inst/examples/FastLM/ImArmadillo.R</a>	2651
<a href="#">inst/examples/FastLM/ImGSL.R</a>	2651
<a href="#">inst/examples/functionCallback/newApiExample.r</a>	2651
<a href="#">inst/examples/Misc/fibonacci.r</a>	2651
<a href="#">inst/examples/Misc/ifelseLooped.r</a>	2651

inst/examples/Misc/newFib.r	2651
inst/examples/Misc/piBySimulation.r	2651
inst/examples/Misc/piSugar.cpp	2651
inst/examples/OpenMP/check.R	2653
inst/examples/OpenMP/OpenMPandInline.r	2653
inst/examples/OpenMP/piWithInterrupts.cpp	2653
inst/examples/performance/extractors.R	2656
inst/examples/performance/performance.R	2656
inst/examples/RcppGibbs/RcppGibbs.R	2656
inst/examples/RcppGibbs/RcppGibbs_Updated.R	2656
inst/examples/RcppGibbs/timeRNGs.R	2656
inst/examples/RcppInline/external_pointer.r	2656
inst/examples/RcppInline/RcppInlineExample.r	2656
inst/examples/RcppInline/RcppInlineWithLibsExamples.r	2656
inst/examples/RcppInline/RcppSimpleExample.r	2656
inst/examples/RcppInline/RObject.r	2656
inst/examples/RcppInline/UncaughtExceptions.r	2656
inst/examples/SugarPerformance/sugarBenchmarks.R	2656
inst/examples/SugarPerformance/Timer.h	2697
inst/examples/SugarPerformance/Timertest.cpp	2656
inst/include/Rcpp.h	2658
inst/include/RcppCommon.h	2976
inst/include/doxygen/Examples.h	2658
inst/include/Rcpp/algo.h	2660
inst/include/Rcpp/algorithm.h	2661
inst/include/Rcpp/as.h	2669
inst/include/Rcpp/barrier.h	2692
inst/include/Rcpp/clone.h	2697
inst/include/Rcpp/complex.h	2698
inst/include/Rcpp/config.h	2701
inst/include/Rcpp/DataFrame.h	2671
inst/include/Rcpp/Dimension.h	2672
inst/include/Rcpp/DottedPair.h	2706
inst/include/Rcpp/DottedPairImpl.h	2673
inst/include/Rcpp/Environment.h	2674
inst/include/Rcpp/exceptions.h	2710
inst/include/Rcpp/exceptions_impl.h	2718
inst/include/Rcpp/Extractor.h	2718
inst/include/Rcpp/Fast.h	2719
inst/include/Rcpp/Formula.h	2719
inst/include/Rcpp/Function.h	2720
inst/include/Rcpp/grow.h	2721
inst/include/Rcpp/InputParameter.h	2724
inst/include/Rcpp/InternalFunction.h	2725
inst/include/Rcpp/InternalFunctionWithStdFunction.h	2726
inst/include/Rcpp/Interrupt.h	2726
inst/include/Rcpp/is.h	2677
inst/include/Rcpp/lang.h	2729
inst/include/Rcpp/Language.h	2732
inst/include/Rcpp/longlong.h	2733
inst/include/Rcpp/Module.h	2680
inst/include/Rcpp/Na_Proxy.h	2771
inst/include/Rcpp/Named.h	2771
inst/include/Rcpp/Nullable.h	2772

inst/include/Rcpp/Pairlist.h	2772
inst/include/Rcpp/print.h	2775
inst/include/Rcpp/Promise.h	2775
inst/include/Rcpp/r_cast.h	2784
inst/include/Rcpp/Reference.h	2785
inst/include/Rcpp/Rmath.h	2786
inst/include/Rcpp/RNGScope.h	2789
inst/include/Rcpp/RObject.h	2789
inst/include/Rcpp/routines.h	2790
inst/include/Rcpp/S4.h	2687
inst/include/Rcpp/sprintf.h	2797
inst/include/Rcpp/StretchyList.h	2688
inst/include/Rcpp/String.h	2823
inst/include/Rcpp/StringTransformer.h	2825
inst/include/Rcpp/Symbol.h	2915
inst/include/Rcpp/unwindProtect.h	2937
inst/include/Rcpp/Vector.h	2691
inst/include/Rcpp/WeakReference.h	2974
inst/include/Rcpp/XPtr.h	2975
inst/include/Rcpp/api/bones/bones.h	2663
inst/include/Rcpp/api/bones/Date.h	2664
inst/include/Rcpp/api/bones/Datetime.h	2666
inst/include/Rcpp/api/bones/wrap_extra_steps.h	2668
inst/include/Rcpp/api/meat/as.h	2669
inst/include/Rcpp/api/meat/DataFrame.h	2671
inst/include/Rcpp/api/meat/Date.h	2665
inst/include/Rcpp/api/meat/Datetime.h	2667
inst/include/Rcpp/api/meat/Dimension.h	2672
inst/include/Rcpp/api/meat/DottedPairImpl.h	2673
inst/include/Rcpp/api/meat/Environment.h	2673
inst/include/Rcpp/api/meat/export.h	2675
inst/include/Rcpp/api/meat/is.h	2676
inst/include/Rcpp/api/meat/meat.h	2678
inst/include/Rcpp/api/meat/message.h	2679
inst/include/Rcpp/api/meat/protection.h	2683
inst/include/Rcpp/api/meat/proxy.h	2684
inst/include/Rcpp/api/meat/Rcpp_eval.h	2686
inst/include/Rcpp/api/meat/S4.h	2687
inst/include/Rcpp/api/meat/StretchyList.h	2688
inst/include/Rcpp/api/meat/Vector.h	2689
inst/include/Rcpp/api/meat/wrap.h	2692
inst/include/Rcpp/api/meat/module/Module.h	2679
inst/include/Rcpp/Benchmark/Timer.h	2696
inst/include/Rcpp/date_datetime/Date.h	2665
inst/include/Rcpp/date_datetime/date_datetime.h	2703
inst/include/Rcpp/date_datetime/Datetime.h	2667
inst/include/Rcpp/date_datetime/newDatetimeVector.h	2704
inst/include/Rcpp/date_datetime/newDateVector.h	2704
inst/include/Rcpp/date_datetime/oldDatetimeVector.h	2705
inst/include/Rcpp/date_datetime/oldDateVector.h	2706
inst/include/Rcpp/exceptions/cpp11/exceptions.h	2707
inst/include/Rcpp/exceptions/cpp98/exceptions.h	2708
inst/include/Rcpp/hash/hash.h	2722
inst/include/Rcpp/hash/IndexHash.h	2722

inst/include/Rcpp/hash/SelfHash.h	2724
inst/include/Rcpp/iostream/Rstreambuf.h	2728
inst/include/Rcpp/macros/cat.hpp	2734
inst/include/Rcpp/macros/config.hpp	2735
inst/include/Rcpp/macros/debug.h	2737
inst/include/Rcpp/macros/dispatch.h	2741
inst/include/Rcpp/macros/interface.h	2743
inst/include/Rcpp/macros/macros.h	2745
inst/include/Rcpp/macros/module.h	2756
inst/include/Rcpp/macros/traits.h	2758
inst/include/Rcpp/macros/unroll.h	2761
inst/include/Rcpp/macros/xp.h	2763
inst/include/Rcpp/module/class.h	2765
inst/include/Rcpp/module/class_Base.h	2765
inst/include/Rcpp/module/CppFunction.h	2766
inst/include/Rcpp/module/get_return_type.h	2766
inst/include/Rcpp/module/Module.h	2680
inst/include/Rcpp/module/Module_Add_Property.h	2767
inst/include/Rcpp/module/Module_Field.h	2769
inst/include/Rcpp/module/Module_Property.h	2770
inst/include/Rcpp/platform/compiler.h	2773
inst/include/Rcpp/platform/solaris.h	2775
inst/include/Rcpp/protection/Armor.h	2776
inst/include/Rcpp/protection/protection.h	2684
inst/include/Rcpp/protection/Shelter.h	2776
inst/include/Rcpp/protection/Shield.h	2777
inst/include/Rcpp/proxy/AttributeProxy.h	2777
inst/include/Rcpp/proxy/Binding.h	2778
inst/include/Rcpp/proxy/DottedPairProxy.h	2778
inst/include/Rcpp/proxy/FieldProxy.h	2779
inst/include/Rcpp/proxy/GenericProxy.h	2779
inst/include/Rcpp/proxy/NamesProxy.h	2780
inst/include/Rcpp/proxy/ProtectedProxy.h	2780
inst/include/Rcpp/proxy/proxy.h	2685
inst/include/Rcpp/proxy/RObjectMethods.h	2781
inst/include/Rcpp/proxy/SlotProxy.h	2781
inst/include/Rcpp/proxy/TagProxy.h	2782
inst/include/Rcpp/r/headers.h	2782
inst/include/Rcpp/stats/beta.h	2798
inst/include/Rcpp/stats/binom.h	2798
inst/include/Rcpp/stats/cauchy.h	2798
inst/include/Rcpp/stats/chisq.h	2799
inst/include/Rcpp/stats/exp.h	2802
inst/include/Rcpp/stats/f.h	2803
inst/include/Rcpp/stats/gamma.h	2803
inst/include/Rcpp/stats/geom.h	2804
inst/include/Rcpp/stats/hyper.h	2804
inst/include/Rcpp/stats/lnorm.h	2804
inst/include/Rcpp/stats/logis.h	2805
inst/include/Rcpp/stats/nbeta.h	2805
inst/include/Rcpp/stats/nbinom.h	2805
inst/include/Rcpp/stats/nbinom_mu.h	2806
inst/include/Rcpp/stats/nchisq.h	2806
inst/include/Rcpp/stats/nf.h	2806

inst/include/Rcpp/stats/norm.h	2806
inst/include/Rcpp/stats/nt.h	2807
inst/include/Rcpp/stats/pois.h	2807
inst/include/Rcpp/stats/stats.h	2819
inst/include/Rcpp/stats/t.h	2821
inst/include/Rcpp/stats/unif.h	2821
inst/include/Rcpp/stats/weibull.h	2821
inst/include/Rcpp/stats/dpq/dpq.h	2799
inst/include/Rcpp/stats/dpq/macros.h	2748
inst/include/Rcpp/stats/random/random.h	2807
inst/include/Rcpp/stats/random/rbeta.h	2809
inst/include/Rcpp/stats/random/rbinom.h	2809
inst/include/Rcpp/stats/random/rcauchy.h	2810
inst/include/Rcpp/stats/random/rchisq.h	2810
inst/include/Rcpp/stats/random/rexp.h	2811
inst/include/Rcpp/stats/random/rf.h	2811
inst/include/Rcpp/stats/random/rgamma.h	2812
inst/include/Rcpp/stats/random/rgeom.h	2812
inst/include/Rcpp/stats/random/rhyper.h	2813
inst/include/Rcpp/stats/random/rlnorm.h	2813
inst/include/Rcpp/stats/random/rlogis.h	2814
inst/include/Rcpp/stats/random/rnbinom.h	2814
inst/include/Rcpp/stats/random/rnbinom_mu.h	2815
inst/include/Rcpp/stats/random/rnchisq.h	2815
inst/include/Rcpp/stats/random/rnorm.h	2816
inst/include/Rcpp/stats/random/rpois.h	2816
inst/include/Rcpp/stats/random/rsignrank.h	2817
inst/include/Rcpp/stats/random/rt.h	2817
inst/include/Rcpp/stats/random/runif.h	2818
inst/include/Rcpp/stats/random/rweibull.h	2818
inst/include/Rcpp/stats/random/rwilcox.h	2819
inst/include/Rcpp/storage/NoProtectStorage.h	2822
inst/include/Rcpp/storage/PreserveStorage.h	2822
inst/include/Rcpp/storage/storage.h	2823
inst/include/Rcpp/sugar/Range.h	2912
inst/include/Rcpp/sugar/sets.h	2912
inst/include/Rcpp/sugar/sugar.h	2913
inst/include/Rcpp/sugar/sugar_forward.h	2913
inst/include/Rcpp/sugar/undoRmath.h	2914
inst/include/Rcpp/sugar/block/block.h	2826
inst/include/Rcpp/sugar/block/SugarBlock_1.h	2826
inst/include/Rcpp/sugar/block/SugarBlock_2.h	2828
inst/include/Rcpp/sugar/block/SugarBlock_3.h	2829
inst/include/Rcpp/sugar/block/SugarMath.h	2831
inst/include/Rcpp/sugar/block/Vectorized_Math.h	2832
inst/include/Rcpp/sugar/functions/all.h	2833
inst/include/Rcpp/sugar/functions/any.h	2834
inst/include/Rcpp/sugar/functions/cbind.h	2835
inst/include/Rcpp/sugar/functions/clamp.h	2844
inst/include/Rcpp/sugar/functions/complex.h	2700
inst/include/Rcpp/sugar/functions/cummax.h	2845
inst/include/Rcpp/sugar/functions/cummin.h	2845
inst/include/Rcpp/sugar/functions/cumprod.h	2846
inst/include/Rcpp/sugar/functions/cumsum.h	2846

inst/include/Rcpp/sugar/functions/diff.h	2847
inst/include/Rcpp/sugar/functions/duplicated.h	2848
inst/include/Rcpp/sugar/functions/functions.h	2848
inst/include/Rcpp/sugar/functions/head.h	2849
inst/include/Rcpp/sugar/functions/ifelse.h	2850
inst/include/Rcpp/sugar/functions/is_finite.h	2851
inst/include/Rcpp/sugar/functions/is_infinite.h	2852
inst/include/Rcpp/sugar/functions/is_na.h	2853
inst/include/Rcpp/sugar/functions/is_nan.h	2854
inst/include/Rcpp/sugar/functions/lapply.h	2856
inst/include/Rcpp/sugar/functions/Lazy.h	2856
inst/include/Rcpp/sugar/functions/mapply.h	2857
inst/include/Rcpp/sugar/functions/match.h	2859
inst/include/Rcpp/sugar/functions/math.h	2859
inst/include/Rcpp/sugar/functions/max.h	2860
inst/include/Rcpp/sugar/functions/mean.h	2860
inst/include/Rcpp/sugar/functions/median.h	2861
inst/include/Rcpp/sugar/functions/min.h	2862
inst/include/Rcpp/sugar/functions/na_omit.h	2863
inst/include/Rcpp/sugar/functions/pmax.h	2863
inst/include/Rcpp/sugar/functions/pmin.h	2864
inst/include/Rcpp/sugar/functions/pow.h	2865
inst/include/Rcpp/sugar/functions/range.h	2865
inst/include/Rcpp/sugar/functions/rep.h	2866
inst/include/Rcpp/sugar/functions/rep_each.h	2867
inst/include/Rcpp/sugar/functions/rep_len.h	2867
inst/include/Rcpp/sugar/functions/rev.h	2868
inst/include/Rcpp/sugar/functions/rowSums.h	2868
inst/include/Rcpp/sugar/functions/sample.h	2872
inst/include/Rcpp/sugar/functions/sapply.h	2873
inst/include/Rcpp/sugar/functions/sd.h	2874
inst/include/Rcpp/sugar/functions/self_match.h	2874
inst/include/Rcpp/sugar/functions/seq_along.h	2875
inst/include/Rcpp/sugar/functions/setdiff.h	2876
inst/include/Rcpp/sugar/functions/sign.h	2876
inst/include/Rcpp/sugar/functions/sum.h	2879
inst/include/Rcpp/sugar/functions/table.h	2880
inst/include/Rcpp/sugar/functions/tail.h	2881
inst/include/Rcpp/sugar/functions/unique.h	2881
inst/include/Rcpp/sugar/functions/var.h	2882
inst/include/Rcpp/sugar/functions/which_max.h	2883
inst/include/Rcpp/sugar/functions/which_min.h	2883
inst/include/Rcpp/sugar/functions/mapply/mapply_2.h	2857
inst/include/Rcpp/sugar/functions/mapply/mapply_3.h	2858
inst/include/Rcpp/sugar/functions/strings/collapse.h	2877
inst/include/Rcpp/sugar/functions/strings/strings.h	2878
inst/include/Rcpp/sugar/functions/strings/trimws.h	2878
inst/include/Rcpp/sugar/logical/and.h	2884
inst/include/Rcpp/sugar/logical/can_have_na.h	2886
inst/include/Rcpp/sugar/logical/is.h	2678
inst/include/Rcpp/sugar/logical/logical.h	2886
inst/include/Rcpp/sugar/logical/not.h	2887
inst/include/Rcpp/sugar/logical/or.h	2889
inst/include/Rcpp/sugar/logical/SingleLogicalResult.h	2891

inst/include/Rcpp/sugar/matrix/as_vector.h	2891
inst/include/Rcpp/sugar/matrix/col.h	2892
inst/include/Rcpp/sugar/matrix/diag.h	2892
inst/include/Rcpp/sugar/matrix/lower_tri.h	2893
inst/include/Rcpp/sugar/matrix/matrix_functions.h	2893
inst/include/Rcpp/sugar/matrix/outer.h	2894
inst/include/Rcpp/sugar/matrix/row.h	2894
inst/include/Rcpp/sugar/matrix/tools.h	2895
inst/include/Rcpp/sugar/matrix/upper_tri.h	2896
inst/include/Rcpp/sugar/nona/nona.h	2896
inst/include/Rcpp/sugar/operators/Comparator.h	2897
inst/include/Rcpp/sugar/operators/Comparator_With_One_Value.h	2897
inst/include/Rcpp/sugar/operators/divides.h	2898
inst/include/Rcpp/sugar/operators/logical_operators__Vector__primitive.h	2899
inst/include/Rcpp/sugar/operators/logical_operators__Vector__Vector.h	2903
inst/include/Rcpp/sugar/operators/minus.h	2906
inst/include/Rcpp/sugar/operators/not.h	2888
inst/include/Rcpp/sugar/operators/operators.h	2907
inst/include/Rcpp/sugar/operators/plus.h	2907
inst/include/Rcpp/sugar/operators/r_binary_op.h	2908
inst/include/Rcpp/sugar/operators/times.h	2909
inst/include/Rcpp/sugar/operators/unary_minus.h	2911
inst/include/Rcpp/sugar/tools/iterator.h	2914
inst/include/Rcpp/traits/char_type.h	2915
inst/include/Rcpp/traits/enable_if.h	2916
inst/include/Rcpp/traits/expands_to_logical.h	2916
inst/include/Rcpp/traits/get_na.h	2917
inst/include/Rcpp/traits/has_iterator.h	2917
inst/include/Rcpp/traits/has_na.h	2918
inst/include/Rcpp/traits/if_h	2919
inst/include/Rcpp/traits/init_type.h	2919
inst/include/Rcpp/traits/integral_constant.h	2920
inst/include/Rcpp/traits/is_arithmetic.h	2920
inst/include/Rcpp/traits/is_bool.h	2921
inst/include/Rcpp/traits/is_const.h	2922
inst/include/Rcpp/traits/is_convertible.h	2922
inst/include/Rcpp/traits/is_eigen_base.h	2923
inst/include/Rcpp/traits/is_finite.h	2851
inst/include/Rcpp/traits/is_infinite.h	2853
inst/include/Rcpp/traits/is_module_object.h	2923
inst/include/Rcpp/traits/is_na.h	2854
inst/include/Rcpp/traits/is_nan.h	2855
inst/include/Rcpp/traits/is_pointer.h	2924
inst/include/Rcpp/traits/is_primitive.h	2924
inst/include/Rcpp/traits/is_reference.h	2925
inst/include/Rcpp/traits/is_sugar_expression.h	2925
inst/include/Rcpp/traits/is_trivial.h	2926
inst/include/Rcpp/traits/is_wide_string.h	2926
inst/include/Rcpp/traits/longlong.h	2733
inst/include/Rcpp/traits/matrix_interface.h	2927
inst/include/Rcpp/traits/module_wrap_traits.h	2927
inst/include/Rcpp/traits/named_object.h	2928
inst/include/Rcpp/traits/num2type.h	2929
inst/include/Rcpp/traits/one_type.h	2929



<a href="#">inst/include/Rcpp/traits/r_sexptype_traits.h</a>	2929
<a href="#">inst/include/Rcpp/traits/r_type_traits.h</a>	2930
<a href="#">inst/include/Rcpp/traits/remove_const.h</a>	2932
<a href="#">inst/include/Rcpp/traits/remove_const_and_reference.h</a>	2933
<a href="#">inst/include/Rcpp/traits/remove_reference.h</a>	2933
<a href="#">inst/include/Rcpp/traits/result_of.h</a>	2934
<a href="#">inst/include/Rcpp/traits/same_type.h</a>	2934
<a href="#">inst/include/Rcpp/traits/storage_type.h</a>	2935
<a href="#">inst/include/Rcpp/traits/traits.h</a>	2759
<a href="#">inst/include/Rcpp/traits/un_pointer.h</a>	2935
<a href="#">inst/include/Rcpp/traits/wrap_type_traits.h</a>	2936
<a href="#">inst/include/Rcpp/utils/tinyformat.h</a>	2957
<a href="#">inst/include/Rcpp/utils/tinyformat/tinyformat.h</a>	2938
<a href="#">inst/include/Rcpp/vector/00_forward_proxy.h</a>	2958
<a href="#">inst/include/Rcpp/vector/00_forward_Vector.h</a>	2959
<a href="#">inst/include/Rcpp/vector/ChildVector.h</a>	2959
<a href="#">inst/include/Rcpp/vector/const_generic_proxy.h</a>	2959
<a href="#">inst/include/Rcpp/vector/const_string_proxy.h</a>	2960
<a href="#">inst/include/Rcpp/vector/converter.h</a>	2961
<a href="#">inst/include/Rcpp/vector/DimNameProxy.h</a>	2961
<a href="#">inst/include/Rcpp/vector/generic_proxy.h</a>	2962
<a href="#">inst/include/Rcpp/vector/instantiation.h</a>	2962
<a href="#">inst/include/Rcpp/vector/LazyVector.h</a>	2963
<a href="#">inst/include/Rcpp/vector/ListOf.h</a>	2964
<a href="#">inst/include/Rcpp/vector/Matrix.h</a>	2964
<a href="#">inst/include/Rcpp/vector/MatrixBase.h</a>	2966
<a href="#">inst/include/Rcpp/vector/MatrixColumn.h</a>	2967
<a href="#">inst/include/Rcpp/vector/MatrixRow.h</a>	2967
<a href="#">inst/include/Rcpp/vector/no_init.h</a>	2968
<a href="#">inst/include/Rcpp/vector/proxy.h</a>	2685
<a href="#">inst/include/Rcpp/vector/RangeIndexer.h</a>	2968
<a href="#">inst/include/Rcpp/vector/string_proxy.h</a>	2970
<a href="#">inst/include/Rcpp/vector/SubMatrix.h</a>	2970
<a href="#">inst/include/Rcpp/vector/Subsetter.h</a>	2971
<a href="#">inst/include/Rcpp/vector/swap.h</a>	2972
<a href="#">inst/include/Rcpp/vector/traits.h</a>	2761
<a href="#">inst/include/Rcpp/vector/Vector.h</a>	2690
<a href="#">inst/include/Rcpp/vector/vector_from_string.h</a>	2973
<a href="#">inst/include/Rcpp/vector/VectorBase.h</a>	2974
<a href="#">src/api.cpp</a>	2978
<a href="#">src/attributes.cpp</a>	2981
<a href="#">src/barrier.cpp</a>	2985
<a href="#">src/date.cpp</a>	2996
<a href="#">src/internal.h</a>	3011
<a href="#">src/module.cpp</a>	3022
<a href="#">src/rcpp_init.cpp</a>	3033



## Chapter 5

# Namespace Documentation

### 5.1 R Namespace Reference

#### Functions

- double [norm\\_rand](#) (void)
- double [unif\\_rand](#) (void)
- double [exp\\_rand](#) (void)
- double [dnorm](#) (double x, double mu, double sigma, int lg)
- double [pnorm](#) (double x, double mu, double sigma, int lt, int lg)
- double [qnorm](#) (double p, double mu, double sigma, int lt, int lg)
- double [rnorm](#) (double mu, double sigma)
- void [pnorm\\_both](#) (double x, double \*cum, double \*ccum, int lt, int lg)
- double [dunif](#) (double x, double a, double b, int lg)
- double [punif](#) (double x, double a, double b, int lt, int lg)
- double [qunif](#) (double p, double a, double b, int lt, int lg)
- double [runif](#) (double a, double b)
- double [dgamma](#) (double x, double shp, double scl, int lg)
- double [pgamma](#) (double x, double alp, double scl, int lt, int lg)
- double [qgamma](#) (double p, double alp, double scl, int lt, int lg)
- double [rgamma](#) (double a, double scl)
- double [log1pmx](#) (double x)
- double [log1pexp](#) (double x)
- double [lgamma1p](#) (double a)
- double [logspace\\_add](#) (double lx, double ly)
- double [logspace\\_sub](#) (double lx, double ly)
- double [dbeta](#) (double x, double a, double b, int lg)
- double [pbeta](#) (double x, double p, double q, int lt, int lg)
- double [qbeta](#) (double a, double p, double q, int lt, int lg)
- double [rbeta](#) (double a, double b)
- double [dlnorm](#) (double x, double ml, double sl, int lg)
- double [plnorm](#) (double x, double ml, double sl, int lt, int lg)
- double [qlnorm](#) (double p, double ml, double sl, int lt, int lg)
- double [rlnorm](#) (double ml, double sl)

- double [dchisq](#) (double x, double df, int lg)
- double [pchisq](#) (double x, double df, int lt, int lg)
- double [qchisq](#) (double p, double df, int lt, int lg)
- double [rchisq](#) (double df)
- double [dnchisq](#) (double x, double df, double ncp, int lg)
- double [pnchisq](#) (double x, double df, double ncp, int lt, int lg)
- double [qnchisq](#) (double p, double df, double ncp, int lt, int lg)
- double [rnchisq](#) (double df, double lb)
- double [df](#) (double x, double df1, double df2, int lg)
- double [pf](#) (double x, double df1, double df2, int lt, int lg)
- double [qf](#) (double p, double df1, double df2, int lt, int lg)
- double [rf](#) (double df1, double df2)
- double [dt](#) (double x, double n, int lg)
- double [pt](#) (double x, double n, int lt, int lg)
- double [qt](#) (double p, double n, int lt, int lg)
- double [rt](#) (double n)
- double [dbinom](#) (double x, double n, double p, int lg)
- double [pbinom](#) (double x, double n, double p, int lt, int lg)
- double [qbinom](#) (double p, double n, double m, int lt, int lg)
- double [rbinom](#) (double n, double p)
- void [rmultinom](#) (int n, double \*prob, int k, int \*rn)
- double [dcauchy](#) (double x, double lc, double sl, int lg)
- double [pcauchy](#) (double x, double lc, double sl, int lt, int lg)
- double [qcauchy](#) (double p, double lc, double sl, int lt, int lg)
- double [rcauchy](#) (double lc, double sl)
- double [dexp](#) (double x, double sl, int lg)
- double [pexp](#) (double x, double sl, int lt, int lg)
- double [qexp](#) (double p, double sl, int lt, int lg)
- double [rexp](#) (double sl)
- double [dgeom](#) (double x, double p, int lg)
- double [pgeom](#) (double x, double p, int lt, int lg)
- double [qgeom](#) (double p, double pb, int lt, int lg)
- double [rgeom](#) (double p)
- double [dhyper](#) (double x, double r, double b, double n, int lg)
- double [phyper](#) (double x, double r, double b, double n, int lt, int lg)
- double [qhyper](#) (double p, double r, double b, double n, int lt, int lg)
- double [rhyper](#) (double r, double b, double n)
- double [dnbinom](#) (double x, double sz, double pb, int lg)
- double [pnbinom](#) (double x, double sz, double pb, int lt, int lg)
- double [qnbinom](#) (double p, double sz, double pb, int lt, int lg)
- double [rnbinom](#) (double sz, double pb)
- double [dnbinom\\_mu](#) (double x, double sz, double mu, int lg)
- double [pnbinom\\_mu](#) (double x, double sz, double mu, int lt, int lg)
- double [qnbinom\\_mu](#) (double x, double sz, double mu, int lt, int lg)
- double [dpois](#) (double x, double lb, int lg)
- double [ppois](#) (double x, double lb, int lt, int lg)
- double [qpois](#) (double p, double lb, int lt, int lg)
- double [rpois](#) (double mu)
- double [dweibull](#) (double x, double sh, double sl, int lg)
- double [pweibull](#) (double x, double sh, double sl, int lt, int lg)
- double [qweibull](#) (double p, double sh, double sl, int lt, int lg)

- double [rweibull](#) (double sh, double sl)
- double [dlogis](#) (double x, double lc, double sl, int lg)
- double [plogis](#) (double x, double lc, double sl, int lt, int lg)
- double [qlogis](#) (double p, double lc, double sl, int lt, int lg)
- double [rlogis](#) (double lc, double sl)
- double [dnbeta](#) (double x, double a, double b, double ncp, int lg)
- double [pnbeta](#) (double x, double a, double b, double ncp, int lt, int lg)
- double [qnbeta](#) (double p, double a, double b, double ncp, int lt, int lg)
- double [rnbeta](#) (double a, double b, double np)
- double [dnf](#) (double x, double df1, double df2, double ncp, int lg)
- double [pnf](#) (double x, double df1, double df2, double ncp, int lt, int lg)
- double [qnf](#) (double p, double df1, double df2, double ncp, int lt, int lg)
- double [dnt](#) (double x, double df, double ncp, int lg)
- double [pnt](#) (double x, double df, double ncp, int lt, int lg)
- double [qnt](#) (double p, double df, double ncp, int lt, int lg)
- double [ptukey](#) (double q, double rr, double cc, double df, int lt, int lg)
- double [qtukey](#) (double p, double rr, double cc, double df, int lt, int lg)
- double [dwilcox](#) (double x, double m, double n, int lg)
- double [pwilcox](#) (double q, double m, double n, int lt, int lg)
- double [qwilcox](#) (double x, double m, double n, int lt, int lg)
- double [rwilcox](#) (double m, double n)
- double [dsignrank](#) (double x, double n, int lg)
- double [psignrank](#) (double x, double n, int lt, int lg)
- double [qsignrank](#) (double x, double n, int lt, int lg)
- double [rsignrank](#) (double n)
- double [gammafn](#) (double x)
- double [lgammafn](#) (double x)
- double [lgammafn\\_sign](#) (double x, int \*sgn)
- void [dpsifn](#) (double x, int n, int kode, int m, double \*ans, int \*nz, int \*ierr)
- double [psigamma](#) (double x, double deriv)
- double [digamma](#) (double x)
- double [trigamma](#) (double x)
- double [tetragamma](#) (double x)
- double [pentagamma](#) (double x)
- double [beta](#) (double a, double b)
- double [lbeta](#) (double a, double b)
- double [choose](#) (double n, double k)
- double [lchoose](#) (double n, double k)
- double [bessel\\_i](#) (double x, double al, double ex)
- double [bessel\\_j](#) (double x, double al)
- double [bessel\\_k](#) (double x, double al, double ex)
- double [bessel\\_y](#) (double x, double al)
- double [bessel\\_i\\_ex](#) (double x, double al, double ex, double \*bi)
- double [bessel\\_j\\_ex](#) (double x, double al, double \*bj)
- double [bessel\\_k\\_ex](#) (double x, double al, double ex, double \*bk)
- double [bessel\\_y\\_ex](#) (double x, double al, double \*by)
- double [hypot](#) (double a, double b)
- double [expm1](#) (double x)
- double [log1p](#) (double x)
- int [imax2](#) (int x, int y)
- int [imin2](#) (int x, int y)

- double [fmax2](#) (double x, double y)
- double [fmin2](#) (double x, double y)
- double [sign](#) (double x)
- double [fprec](#) (double x, double dg)
- double [fround](#) (double x, double dg)
- double [fsign](#) (double x, double y)
- double [ftrunc](#) (double x)

## 5.1.1 Function Documentation

### 5.1.1.1 [bessel\\_i\(\)](#)

```
double R::bessel_i (  
    double x,  
    double a1,  
    double ex ) [inline]
```

Definition at line 209 of file Rmath.h.

### 5.1.1.2 [bessel\\_i\\_ex\(\)](#)

```
double R::bessel_i_ex (  
    double x,  
    double a1,  
    double ex,  
    double * bi ) [inline]
```

Definition at line 213 of file Rmath.h.

### 5.1.1.3 [bessel\\_j\(\)](#)

```
double R::bessel_j (  
    double x,  
    double a1 ) [inline]
```

Definition at line 210 of file Rmath.h.

#### 5.1.1.4 `bessel_j_ex()`

```
double R::bessel_j_ex (  
    double x,  
    double a1,  
    double * bj ) [inline]
```

Definition at line 214 of file Rmath.h.

#### 5.1.1.5 `bessel_k()`

```
double R::bessel_k (  
    double x,  
    double a1,  
    double ex ) [inline]
```

Definition at line 211 of file Rmath.h.

#### 5.1.1.6 `bessel_k_ex()`

```
double R::bessel_k_ex (  
    double x,  
    double a1,  
    double ex,  
    double * bk ) [inline]
```

Definition at line 215 of file Rmath.h.

#### 5.1.1.7 `bessel_y()`

```
double R::bessel_y (  
    double x,  
    double a1 ) [inline]
```

Definition at line 212 of file Rmath.h.

#### 5.1.1.8 `bessel_y_ex()`

```
double R::bessel_y_ex (  
    double x,  
    double a1,  
    double * by ) [inline]
```

Definition at line 216 of file Rmath.h.

#### 5.1.1.9 `beta()`

```
double R::beta (  
    double a,  
    double b ) [inline]
```

Definition at line 202 of file Rmath.h.

#### 5.1.1.10 `choose()`

```
double R::choose (  
    double n,  
    double k ) [inline]
```

Definition at line 205 of file Rmath.h.

#### 5.1.1.11 `dbeta()`

```
double R::dbeta (  
    double x,  
    double a,  
    double b,  
    int lg ) [inline]
```

Definition at line 60 of file Rmath.h.

### 5.1.1.12 `dbinom()`

```
double R::dbinom (
    double x,
    double n,
    double p,
    int lg ) [inline]
```

Definition at line 96 of file Rmath.h.

### 5.1.1.13 `dcauchy()`

```
double R::dcauchy (
    double x,
    double lc,
    double sl,
    int lg ) [inline]
```

Definition at line 105 of file Rmath.h.

### 5.1.1.14 `dchisq()`

```
double R::dchisq (
    double x,
    double df,
    int lg ) [inline]
```

Definition at line 72 of file Rmath.h.

References `df()`.

Here is the call graph for this function:



#### 5.1.1.15 `dexp()`

```
double R::dexp (
    double x,
    double sl,
    int lg ) [inline]
```

Definition at line 111 of file Rmath.h.

#### 5.1.1.16 `df()`

```
double R::df (
    double x,
    double df1,
    double df2,
    int lg ) [inline]
```

Definition at line 84 of file Rmath.h.

Referenced by `dchisq()`, `dunchisq()`, `dnt()`, `Rcpp::internal::empty_data_frame()`, `pchisq()`, `pnchisq()`, `pnt()`, `ptukey()`, `qchisq()`, `qnchisq()`, `qnt()`, `qtukey()`, `rchisq()`, `Rcpp::rchisq()`, `rnchisq()`, `Rcpp::rnchisq()`, and `Rcpp::rt()`.

#### 5.1.1.17 `dgamma()`

```
double R::dgamma (
    double x,
    double shp,
    double scl,
    int lg ) [inline]
```

Definition at line 48 of file Rmath.h.

#### 5.1.1.18 `dgeom()`

```
double R::dgeom (
    double x,
    double p,
    int lg ) [inline]
```

Definition at line 117 of file Rmath.h.



### 5.1.1.19 dhyper()

```
double R::dhyper (
    double x,
    double r,
    double b,
    double n,
    int lg ) [inline]
```

Definition at line 123 of file Rmath.h.

### 5.1.1.20 digamma()

```
double R::digamma (
    double x ) [inline]
```

Definition at line 197 of file Rmath.h.

### 5.1.1.21 dlnorm()

```
double R::dlnorm (
    double x,
    double ml,
    double sl,
    int lg ) [inline]
```

Definition at line 66 of file Rmath.h.

### 5.1.1.22 dlogis()

```
double R::dlogis (
    double x,
    double lc,
    double sl,
    int lg ) [inline]
```

Definition at line 154 of file Rmath.h.

### 5.1.1.23 `dnbeta()`

```
double R::dnbeta (
    double x,
    double a,
    double b,
    double ncp,
    int lg ) [inline]
```

Definition at line 160 of file `Rmath.h`.

### 5.1.1.24 `dnbinom()`

```
double R::dnbinom (
    double x,
    double sz,
    double pb,
    int lg ) [inline]
```

Definition at line 129 of file `Rmath.h`.

### 5.1.1.25 `dnbinom_mu()`

```
double R::dnbinom_mu (
    double x,
    double sz,
    double mu,
    int lg ) [inline]
```

Definition at line 135 of file `Rmath.h`.

### 5.1.1.26 `dnchisq()`

```
double R::dnchisq (
    double x,
    double df,
    double ncp,
    int lg ) [inline]
```

Definition at line 78 of file `Rmath.h`.

References `df()`.

Here is the call graph for this function:



#### 5.1.1.27 `dnf()`

```
double R::dnf (
    double x,
    double df1,
    double df2,
    double ncp,
    int lg ) [inline]
```

Definition at line 166 of file `Rmath.h`.

#### 5.1.1.28 `dnorm()`

```
double R::dnorm (
    double x,
    double mu,
    double sigma,
    int lg ) [inline]
```

Definition at line 35 of file `Rmath.h`.

### 5.1.1.29 dnt()

```
double R::dnt (
    double x,
    double df,
    double ncp,
    int lg ) [inline]
```

Definition at line 171 of file Rmath.h.

References [df\(\)](#).

Here is the call graph for this function:



### 5.1.1.30 dpois()

```
double R::dpois (
    double x,
    double lb,
    int lg ) [inline]
```

Definition at line 142 of file Rmath.h.

### 5.1.1.31 dpsifn()

```
void R::dpsifn (
    double x,
    int n,
    int kode,
    int m,
    double * ans,
    int * nz,
    int * ierr ) [inline]
```

Definition at line 195 of file Rmath.h.

### 5.1.1.32 `dsignrank()`

```
double R::dsignrank (
    double x,
    double n,
    int lg ) [inline]
```

Definition at line 186 of file Rmath.h.

### 5.1.1.33 `dt()`

```
double R::dt (
    double x,
    double n,
    int lg ) [inline]
```

Definition at line 90 of file Rmath.h.

Referenced by `Rcpp::Date::Date()`, `Rcpp::Datetime::Datetime()`, and `Rcpp::Datetime::update_tm()`.

### 5.1.1.34 `dunif()`

```
double R::dunif (
    double x,
    double a,
    double b,
    int lg ) [inline]
```

Definition at line 42 of file Rmath.h.

### 5.1.1.35 `dweibull()`

```
double R::dweibull (
    double x,
    double sh,
    double sl,
    int lg ) [inline]
```

Definition at line 148 of file Rmath.h.

### 5.1.1.36 dwilcox()

```
double R::dwilcox (
    double x,
    double m,
    double n,
    int lg ) [inline]
```

Definition at line 180 of file Rmath.h.

### 5.1.1.37 exp\_rand()

```
double R::exp_rand (
    void ) [inline]
```

Definition at line 32 of file Rmath.h.

Referenced by `Rcpp::stats::ExpGenerator::operator()()`, `Rcpp::stats::ExpGenerator__rate1::operator()()`, and `Rcpp::stats::GeomGenerator::operator()()`.

### 5.1.1.38 expm1()

```
double R::expm1 (
    double x ) [inline]
```

Referenced by `Rcpp::stats::p_exp_0()`, and `Rcpp::stats::pweibull_1()`.

### 5.1.1.39 fmax2()

```
double R::fmax2 (
    double x,
    double y ) [inline]
```

Definition at line 232 of file Rmath.h.

#### 5.1.1.40 fmin2()

```
double R::fmin2 (
    double x,
    double y ) [inline]
```

Definition at line 233 of file Rmath.h.

#### 5.1.1.41 fprec()

```
double R::fprec (
    double x,
    double dg ) [inline]
```

Definition at line 235 of file Rmath.h.

#### 5.1.1.42 fround()

```
double R::fround (
    double x,
    double dg ) [inline]
```

Definition at line 236 of file Rmath.h.

#### 5.1.1.43 fsign()

```
double R::fsign (
    double x,
    double y ) [inline]
```

Definition at line 237 of file Rmath.h.

#### 5.1.1.44 ftrunc()

```
double R::ftrunc (
    double x ) [inline]
```

Definition at line 238 of file Rmath.h.

#### 5.1.1.45 `gammafn()`

```
double R::gammafn (  
    double x ) [inline]
```

Definition at line 192 of file Rmath.h.

#### 5.1.1.46 `hypot()`

```
double R::hypot (  
    double a,  
    double b ) [inline]
```

Definition at line 220 of file Rmath.h.

Referenced by `Rcpp::internal::complex__asin()`, `Rcpp::internal::complex__log()`, and `Rcpp::internal::complex__sqrt()`.

#### 5.1.1.47 `imax2()`

```
int R::imax2 (  
    int x,  
    int y ) [inline]
```

Definition at line 230 of file Rmath.h.

#### 5.1.1.48 `imin2()`

```
int R::imin2 (  
    int x,  
    int y ) [inline]
```

Definition at line 231 of file Rmath.h.

#### 5.1.1.49 `lbeta()`

```
double R::lbeta (  
    double a,  
    double b ) [inline]
```

Definition at line 203 of file Rmath.h.



#### 5.1.1.50 lchoose()

```
double R::lchoose (
    double n,
    double k ) [inline]
```

Definition at line 206 of file Rmath.h.

#### 5.1.1.51 lgamma1p()

```
double R::lgamma1p (
    double a ) [inline]
```

Definition at line 55 of file Rmath.h.

#### 5.1.1.52 lgammafn()

```
double R::lgammafn (
    double x ) [inline]
```

Definition at line 193 of file Rmath.h.

#### 5.1.1.53 lgammafn\_sign()

```
double R::lgammafn_sign (
    double x,
    int * sgn ) [inline]
```

Definition at line 194 of file Rmath.h.

#### 5.1.1.54 log1p()

```
double R::log1p (
    double x ) [inline]
```

Referenced by `Rcpp::stats::p_exp_0()`, `Rcpp::stats::plogis_0()`, `Rcpp::stats::plogis_1()`, `Rcpp::stats::pweibull_1()`, `Rcpp::stats::qlogis_0()`, and `Rcpp::stats::qlogis_1()`.

### 5.1.1.55 log1pexp()

```
double R::log1pexp (
    double x ) [inline]
```

Definition at line 54 of file Rmath.h.

### 5.1.1.56 log1pmx()

```
double R::log1pmx (
    double x ) [inline]
```

Definition at line 53 of file Rmath.h.

### 5.1.1.57 logspace\_add()

```
double R::logspace_add (
    double lx,
    double ly ) [inline]
```

Definition at line 56 of file Rmath.h.

### 5.1.1.58 logspace\_sub()

```
double R::logspace_sub (
    double lx,
    double ly ) [inline]
```

Definition at line 57 of file Rmath.h.

### 5.1.1.59 norm\_rand()

```
double R::norm_rand (
    void ) [inline]
```

Definition at line 30 of file Rmath.h.

Referenced by `Rcpp::stats::LNormGenerator::operator()()`, `Rcpp::stats::LNormGenerator_1::operator()()`, `Rcpp::stats::LNormGenerator_0::operator()()`, `Rcpp::stats::NormGenerator::operator()()`, `Rcpp::stats::NormGenerator__sd1::operator()()`, `Rcpp::stats::NormGenerator__mean0::operator()()`, `Rcpp::stats::NormGenerator__mean0__sd1::operator()()`, and `Rcpp::stats::TGenerator::operator()()`.

### 5.1.1.60 `pbeta()`

```
double R::pbeta (
    double x,
    double p,
    double q,
    int lt,
    int lg ) [inline]
```

Definition at line 61 of file Rmath.h.

### 5.1.1.61 `pbinom()`

```
double R::pbinom (
    double x,
    double n,
    double p,
    int lt,
    int lg ) [inline]
```

Definition at line 97 of file Rmath.h.

### 5.1.1.62 `pcauchy()`

```
double R::pcauchy (
    double x,
    double lc,
    double sl,
    int lt,
    int lg ) [inline]
```

Definition at line 106 of file Rmath.h.

### 5.1.1.63 pchisq()

```
double R::pchisq (  
    double x,  
    double df,  
    int lt,  
    int lg ) [inline]
```

Definition at line 73 of file Rmath.h.

References [df\(\)](#).

Here is the call graph for this function:



### 5.1.1.64 pentagamma()

```
double R::pentagamma (  
    double x ) [inline]
```

Definition at line 200 of file Rmath.h.

### 5.1.1.65 pexp()

```
double R::pexp (  
    double x,  
    double sl,  
    int lt,  
    int lg ) [inline]
```

Definition at line 112 of file Rmath.h.

### 5.1.1.66 pf()

```
double R::pf (
    double x,
    double df1,
    double df2,
    int lt,
    int lg ) [inline]
```

Definition at line 85 of file Rmath.h.

### 5.1.1.67 pgamma()

```
double R::pgamma (
    double x,
    double alp,
    double scl,
    int lt,
    int lg ) [inline]
```

Definition at line 49 of file Rmath.h.

### 5.1.1.68 pgeom()

```
double R::pgeom (
    double x,
    double p,
    int lt,
    int lg ) [inline]
```

Definition at line 118 of file Rmath.h.

### 5.1.1.69 phyper()

```
double R::phyper (
    double x,
    double r,
    double b,
    double n,
    int lt,
    int lg ) [inline]
```

Definition at line 124 of file Rmath.h.

### 5.1.1.70 `pnorm()`

```
double R::pnorm (
    double x,
    double ml,
    double sl,
    int lt,
    int lg ) [inline]
```

Definition at line 67 of file `Rmath.h`.

### 5.1.1.71 `plogis()`

```
double R::plogis (
    double x,
    double lc,
    double sl,
    int lt,
    int lg ) [inline]
```

Definition at line 155 of file `Rmath.h`.

### 5.1.1.72 `pnbeta()`

```
double R::pnbeta (
    double x,
    double a,
    double b,
    double ncp,
    int lt,
    int lg ) [inline]
```

Definition at line 161 of file `Rmath.h`.

### 5.1.1.73 `pnbinom()`

```
double R::pnbinom (
    double x,
    double sz,
    double pb,
    int lt,
    int lg ) [inline]
```

Definition at line 130 of file `Rmath.h`.

### 5.1.1.74 `pnbinom_mu()`

```
double R::pnbinom_mu (  
    double x,  
    double sz,  
    double mu,  
    int lt,  
    int lg ) [inline]
```

Definition at line 136 of file Rmath.h.

### 5.1.1.75 `pnchisq()`

```
double R::pnchisq (  
    double x,  
    double df,  
    double ncp,  
    int lt,  
    int lg ) [inline]
```

Definition at line 79 of file Rmath.h.

References `df()`.

Here is the call graph for this function:



### 5.1.1.76 `pnf()`

```
double R::pnf (  
    double x,  
    double df1,  
    double df2,  
    double ncp,  
    int lt,  
    int lg ) [inline]
```

Definition at line 167 of file Rmath.h.

### 5.1.1.77 pnorm()

```
double R::pnorm (
    double x,
    double mu,
    double sigma,
    int lt,
    int lg ) [inline]
```

Definition at line 36 of file Rmath.h.

### 5.1.1.78 pnorm\_both()

```
void R::pnorm_both (
    double x,
    double * cum,
    double * ccum,
    int lt,
    int lg ) [inline]
```

Definition at line 39 of file Rmath.h.

### 5.1.1.79 pnt()

```
double R::pnt (
    double x,
    double df,
    double ncp,
    int lt,
    int lg ) [inline]
```

Definition at line 172 of file Rmath.h.

References [df\(\)](#).

Here is the call graph for this function:





### 5.1.1.80 ppois()

```
double R::ppois (
    double x,
    double lb,
    int lt,
    int lg ) [inline]
```

Definition at line 143 of file Rmath.h.

### 5.1.1.81 psigamma()

```
double R::psigamma (
    double x,
    double deriv ) [inline]
```

Definition at line 196 of file Rmath.h.

### 5.1.1.82 psignrank()

```
double R::psignrank (
    double x,
    double n,
    int lt,
    int lg ) [inline]
```

Definition at line 187 of file Rmath.h.

### 5.1.1.83 pt()

```
double R::pt (
    double x,
    double n,
    int lt,
    int lg ) [inline]
```

Definition at line 91 of file Rmath.h.

#### 5.1.1.84 ptukey()

```
double R::ptukey (
    double q,
    double rr,
    double cc,
    double dF,
    int lt,
    int lg ) [inline]
```

Definition at line 176 of file Rmath.h.

References [df\(\)](#).

Here is the call graph for this function:



#### 5.1.1.85 punif()

```
double R::punif (
    double x,
    double a,
    double b,
    int lt,
    int lg ) [inline]
```

Definition at line 43 of file Rmath.h.

#### 5.1.1.86 pweibull()

```
double R::pweibull (
    double x,
    double sh,
    double sl,
    int lt,
    int lg ) [inline]
```

Definition at line 149 of file Rmath.h.

**5.1.1.87 pwilcox()**

```
double R::pwilcox (
    double q,
    double m,
    double n,
    int lt,
    int lg ) [inline]
```

Definition at line 181 of file Rmath.h.

**5.1.1.88 qbeta()**

```
double R::qbeta (
    double a,
    double p,
    double q,
    int lt,
    int lg ) [inline]
```

Definition at line 62 of file Rmath.h.

**5.1.1.89 qbinom()**

```
double R::qbinom (
    double p,
    double n,
    double m,
    int lt,
    int lg ) [inline]
```

Definition at line 98 of file Rmath.h.

**5.1.1.90 qcauchy()**

```
double R::qcauchy (
    double p,
    double lc,
    double sl,
    int lt,
    int lg ) [inline]
```

Definition at line 107 of file Rmath.h.

### 5.1.1.91 qchisq()

```
double R::qchisq (  
    double p,  
    double df,  
    int lt,  
    int lg ) [inline]
```

Definition at line 74 of file Rmath.h.

References [df\(\)](#).

Here is the call graph for this function:



### 5.1.1.92 qexp()

```
double R::qexp (  
    double p,  
    double sl,  
    int lt,  
    int lg ) [inline]
```

Definition at line 113 of file Rmath.h.

### 5.1.1.93 qf()

```
double R::qf (  
    double p,  
    double df1,  
    double df2,  
    int lt,  
    int lg ) [inline]
```

Definition at line 86 of file Rmath.h.

#### 5.1.1.94 qgamma()

```
double R::qgamma (
    double p,
    double alp,
    double scl,
    int lt,
    int lg ) [inline]
```

Definition at line 50 of file Rmath.h.

#### 5.1.1.95 qgeom()

```
double R::qgeom (
    double p,
    double pb,
    int lt,
    int lg ) [inline]
```

Definition at line 119 of file Rmath.h.

#### 5.1.1.96 qhyper()

```
double R::qhyper (
    double p,
    double r,
    double b,
    double n,
    int lt,
    int lg ) [inline]
```

Definition at line 125 of file Rmath.h.

#### 5.1.1.97 qlnorm()

```
double R::qlnorm (
    double p,
    double mL,
    double sL,
    int lt,
    int lg ) [inline]
```

Definition at line 68 of file Rmath.h.

**5.1.1.98 qlogis()**

```
double R::qlogis (
    double p,
    double lc,
    double sl,
    int lt,
    int lg ) [inline]
```

Definition at line 156 of file Rmath.h.

**5.1.1.99 qnbeta()**

```
double R::qnbeta (
    double p,
    double a,
    double b,
    double ncp,
    int lt,
    int lg ) [inline]
```

Definition at line 162 of file Rmath.h.

**5.1.1.100 qnbinom()**

```
double R::qnbinom (
    double p,
    double sz,
    double pb,
    int lt,
    int lg ) [inline]
```

Definition at line 131 of file Rmath.h.

**5.1.1.101 qnbinom\_mu()**

```
double R::qnbinom_mu (
    double x,
    double sz,
    double mu,
    int lt,
    int lg ) [inline]
```

Definition at line 137 of file Rmath.h.

### 5.1.1.102 qnchisq()

```
double R::qnchisq (  
    double p,  
    double df,  
    double ncp,  
    int lt,  
    int lg ) [inline]
```

Definition at line 80 of file Rmath.h.

References [df\(\)](#).

Here is the call graph for this function:



### 5.1.1.103 qnf()

```
double R::qnf (  
    double p,  
    double df1,  
    double df2,  
    double ncp,  
    int lt,  
    int lg ) [inline]
```

Definition at line 168 of file Rmath.h.

### 5.1.1.104 qnorm()

```
double R::qnorm (  
    double p,  
    double mu,  
    double sigma,  
    int lt,  
    int lg ) [inline]
```

Definition at line 37 of file Rmath.h.

### 5.1.1.105 qnt()

```
double R::qnt (
    double p,
    double df,
    double ncp,
    int lt,
    int lg ) [inline]
```

Definition at line 173 of file Rmath.h.

References [df\(\)](#).

Here is the call graph for this function:



### 5.1.1.106 qpois()

```
double R::qpois (
    double p,
    double lb,
    int lt,
    int lg ) [inline]
```

Definition at line 144 of file Rmath.h.

### 5.1.1.107 qsignrank()

```
double R::qsignrank (
    double x,
    double n,
    int lt,
    int lg ) [inline]
```

Definition at line 188 of file Rmath.h.



**5.1.1.108 qt()**

```
double R::qt (
    double p,
    double n,
    int lt,
    int lg ) [inline]
```

Definition at line 92 of file Rmath.h.

**5.1.1.109 qtukey()**

```
double R::qtukey (
    double p,
    double rr,
    double cc,
    double df,
    int lt,
    int lg ) [inline]
```

Definition at line 177 of file Rmath.h.

References [df\(\)](#).

Here is the call graph for this function:

**5.1.1.110 qunif()**

```
double R::qunif (
    double p,
    double a,
    double b,
    int lt,
    int lg ) [inline]
```

Definition at line 44 of file Rmath.h.

**5.1.1.111 qweibull()**

```
double R::qweibull (
    double p,
    double sh,
    double sl,
    int lt,
    int lg ) [inline]
```

Definition at line 150 of file Rmath.h.

**5.1.1.112 qwilcox()**

```
double R::qwilcox (
    double x,
    double m,
    double n,
    int lt,
    int lg ) [inline]
```

Definition at line 182 of file Rmath.h.

**5.1.1.113 rbeta()**

```
double R::rbeta (
    double a,
    double b ) [inline]
```

Definition at line 63 of file Rmath.h.

**5.1.1.114 rbinom()**

```
double R::rbinom (
    double n,
    double p ) [inline]
```

Definition at line 99 of file Rmath.h.

**5.1.1.115 rcauchy()**

```
double R::rcauchy (
    double lc,
    double sl ) [inline]
```

Definition at line 108 of file Rmath.h.

**5.1.1.116 rchisq()**

```
double R::rchisq (
    double df ) [inline]
```

Definition at line 75 of file Rmath.h.

References [df\(\)](#).

Here is the call graph for this function:

**5.1.1.117 rexp()**

```
double R::rexp (
    double sl ) [inline]
```

Definition at line 114 of file Rmath.h.

**5.1.1.118 rf()**

```
double R::rf (
    double df1,
    double df2 ) [inline]
```

Definition at line 87 of file Rmath.h.

**5.1.1.119 rgamma()**

```
double R::rgamma (
    double a,
    double scl ) [inline]
```

Definition at line 51 of file Rmath.h.

**5.1.1.120 rgeom()**

```
double R::rgeom (
    double p ) [inline]
```

Definition at line 120 of file Rmath.h.

**5.1.1.121 rhyper()**

```
double R::rhyper (
    double r,
    double b,
    double n ) [inline]
```

Definition at line 126 of file Rmath.h.

**5.1.1.122 rlnorm()**

```
double R::rlnorm (
    double ml,
    double sl ) [inline]
```

Definition at line 69 of file Rmath.h.

**5.1.1.123 rlogis()**

```
double R::rlogis (
    double lc,
    double sl ) [inline]
```

Definition at line 157 of file Rmath.h.

**5.1.1.124 rmultinom()**

```
void R::rmultinom (
    int n,
    double * prob,
    int k,
    int * rn ) [inline]
```

Definition at line 102 of file Rmath.h.

**5.1.1.125 rnbeta()**

```
double R::rnbeta (
    double a,
    double b,
    double np ) [inline]
```

Definition at line 163 of file Rmath.h.

**5.1.1.126 rnbinom()**

```
double R::rnbinom (
    double sz,
    double pb ) [inline]
```

Definition at line 132 of file Rmath.h.

**5.1.1.127 rnchisq()**

```
double R::rnchisq (
    double df,
    double lb ) [inline]
```

Definition at line 81 of file Rmath.h.

References [df\(\)](#).

Here is the call graph for this function:



**5.1.1.128 rnorm()**

```
double R::rnorm (
    double mu,
    double sigma ) [inline]
```

Definition at line 38 of file Rmath.h.

**5.1.1.129 rpois()**

```
double R::rpois (
    double mu ) [inline]
```

Definition at line 145 of file Rmath.h.

**5.1.1.130 rsignrank()**

```
double R::rsignrank (
    double n ) [inline]
```

Definition at line 189 of file Rmath.h.

**5.1.1.131 rt()**

```
double R::rt (
    double n ) [inline]
```

Definition at line 93 of file Rmath.h.

**5.1.1.132 runif()**

```
double R::runif (
    double a,
    double b ) [inline]
```

Definition at line 45 of file Rmath.h.

**5.1.1.133 rweibull()**

```
double R::rweibull (
    double sh,
    double sl ) [inline]
```

Definition at line 151 of file Rmath.h.

**5.1.1.134 rwilcox()**

```
double R::rwilcox (
    double m,
    double n ) [inline]
```

Definition at line 183 of file Rmath.h.

**5.1.1.135 sign()**

```
double R::sign (
    double x ) [inline]
```

Definition at line 234 of file Rmath.h.

**5.1.1.136 tetragamma()**

```
double R::tetragamma (
    double x ) [inline]
```

Definition at line 199 of file Rmath.h.

**5.1.1.137 trigamma()**

```
double R::trigamma (
    double x ) [inline]
```

Definition at line 198 of file Rmath.h.

### 5.1.1.138 `unif_rand()`

```
double R::unif_rand (
    void ) [inline]
```

Definition at line 31 of file Rmath.h.

Referenced by `Rcpp::sugar::EmpiricalSample()`, `Rcpp::stats::CauchyGenerator::operator()`, `Rcpp::stats::CauchyGenerator_1::operator()`, `Rcpp::stats::CauchyGenerator_0::operator()`, `Rcpp::stats::LogisGenerator::operator()`, `Rcpp::stats::LogisGenerator_1::operator()`, `Rcpp::stats::LogisGenerator_0::operator()`, `Rcpp::stats::UnifGenerator::operator()`, `Rcpp::stats::UnifGenerator_0__1::operator()`, `Rcpp::stats::WeibullGenerator::operator()`, `Rcpp::stats::WeibullGenerator__scale1::operator()`, `Rcpp::sugar::SampleNoReplace()`, `Rcpp::sugar::SampleReplace()`, and `Rcpp::sugar::WalkerSample()`.

## 5.2 Rcpp Namespace Reference

[Rcpp](#) API.

### Namespaces

- [algorithm](#)
- [attributes](#)
- [internal](#)

*internal implementation details*

- [InternalFunctionWithStdFunction](#)
- [stats](#)
- [sugar](#)
- [traits](#)

*traits used to dispatch wrap*

### Classes

- class [Timer](#)
- class [DataFrame\\_Impl](#)
- class [Date](#)
- class [Datetime](#)
- class [newDatetimeVector](#)
- class [newDateVector](#)
- class [oldDatetimeVector](#)
- class [oldDateVector](#)
- class [Dimension](#)
- class [DottedPairImpl](#)
- class [exception](#)
- class [no\\_such\\_env](#)
- class [file\\_io\\_error](#)
- class [file\\_not\\_found](#)



- class [file\\_exists](#)
- struct [LongjumpException](#)
- class [Fast](#)
- class [InputParameter](#)
- class [ReferenceInputParameter](#)
- class [ConstInputParameter](#)
- class [ConstReferenceInputParameter](#)
- class [Rstreambuf](#)
- class [Rostream](#)
- class [fixed\\_call](#)
- class [unary\\_call](#)
- class [binary\\_call](#)
- class [class\\_Base](#)
- class [CppFunctionBase](#)
- class [CppFunction](#)
- struct [void\\_type](#)
- class [Module](#)
- class [result](#)
- class [object](#)
- class [CppMethod](#)
- class [CppInheritedMethod](#)
- class [SignedConstructor](#)
- class [SignedFactory](#)
- class [SignedMethod](#)
- class [S4\\_CppConstructor](#)
- class [S4\\_CppOverloadedMethods](#)
- class [CppMethodProperty](#)
- class [CppInheritedProperty](#)
- class [CppMethodFinalizer](#)
- class [FunctionFinalizer](#)
- class [S4\\_field](#)
- class [enum\\_](#)
- class [CppClass](#)
- class [CppObject](#)
- class [Na\\_Proxy](#)
- class [Argument](#)
- class [Nullable](#)
- class [Armor](#)
- class [Shelter](#)
- class [Shield](#)
- class [AttributeProxyPolicy](#)
- class [BindingPolicy](#)
- class [DottedPairProxyPolicy](#)
- class [FieldProxyPolicy](#)
- struct [GenericProxy](#)
- class [NamesProxyPolicy](#)
- class [ProtectedProxyPolicy](#)
- class [RObjectMethods](#)
- class [SlotProxyPolicy](#)
- class [TagProxyPolicy](#)
- class [RNGScope](#)

- class [SuspendRNGSynchronizationScope](#)
- class [Generator](#)
- class [NoProtectStorage](#)
- class [PreserveStorage](#)
- class [String](#)
- class [StringTransformer](#)
- struct [can\\_have\\_na](#)
- class [Range](#)
- class [Vector](#)
- class [ChildVector](#)
- class [ListOf](#)
- class [Matrix](#)
- class [MatrixBase](#)
- class [MatrixColumn](#)
- class [ConstMatrixColumn](#)
- class [MatrixRow](#)
- class [ConstMatrixRow](#)
- class [no\\_init\\_vector](#)
- class [no\\_init\\_matrix](#)
- class [SubMatrix](#)
- class [SubsetProxy](#)
- class [VectorBase](#)
- class [SingleLogicalResult](#)
- class [XPtr](#)
- struct [tzhead](#)
- struct [ttinfo](#)
- struct [lsinfo](#)
- struct [state](#)
- struct [rule](#)

## Typedefs

- typedef [uint64\\_t](#) [nanotime\\_t](#)
- typedef [DataFrame\\_Impl< PreserveStorage >](#) [DataFrame](#)
- typedef [oldDateVector](#) [DateVector](#)
- typedef [oldDatetimeVector](#) [DatetimeVector](#)
- typedef [DottedPair\\_Impl< PreserveStorage >](#) [DottedPair](#)
- typedef [Environment\\_Impl< PreserveStorage >](#) [Environment](#)
- typedef [Formula\\_Impl< PreserveStorage >](#) [Formula](#)
- typedef [Function\\_Impl< PreserveStorage >](#) [Function](#)
- typedef [InternalFunction\\_Impl< PreserveStorage >](#) [InternalFunction](#)
- typedef [DottedPairProxyPolicy< Language\\_Impl >::const\\_DottedPairProxy](#) [const\\_Proxy](#)
- typedef [Language\\_Impl< PreserveStorage >](#) [Language](#)
- typedef [bool\(\\* ValidConstructor\)](#) ([SEXP \\*](#), [int](#))
- typedef [bool\(\\* ValidMethod\)](#) ([SEXP \\*](#), [int](#))
- typedef [Pairlist\\_Impl< PreserveStorage >](#) [Pairlist](#)
- typedef [Promise\\_Impl< PreserveStorage >](#) [Promise](#)
- typedef [Reference\\_Impl< PreserveStorage >](#) [Reference](#)
- typedef [RObject\\_Impl< PreserveStorage >](#) [RObject](#)
- typedef [S4\\_Impl< PreserveStorage >](#) [S4](#)

- typedef [StretchyList\\_Impl](#)< [PreserveStorage](#) > [StretchyList](#)
- typedef [Symbol\\_Impl](#)< [NoProtectStorage](#) > [Symbol](#)
- typedef [Vector](#)< [CPLXSXP](#) > [ComplexVector](#)
- typedef [Vector](#)< [INTSXP](#) > [IntegerVector](#)
- typedef [Vector](#)< [LGLSXP](#) > [LogicalVector](#)
- typedef [Vector](#)< [REALSXP](#) > [NumericVector](#)
- typedef [Vector](#)< [REALSXP](#) > [DoubleVector](#)
- typedef [Vector](#)< [RAWSXP](#) > [RawVector](#)
- typedef [Vector](#)< [STRSXP](#) > [CharacterVector](#)
- typedef [Vector](#)< [STRSXP](#) > [StringVector](#)
- typedef [Vector](#)< [VECSXP](#) > [GenericVector](#)
- typedef [Vector](#)< [VECSXP](#) > [List](#)
- typedef [Vector](#)< [EXPRSXP](#) > [ExpressionVector](#)
- typedef [Matrix](#)< [CPLXSXP](#) > [ComplexMatrix](#)
- typedef [Matrix](#)< [INTSXP](#) > [IntegerMatrix](#)
- typedef [Matrix](#)< [LGLSXP](#) > [LogicalMatrix](#)
- typedef [Matrix](#)< [REALSXP](#) > [NumericMatrix](#)
- typedef [Matrix](#)< [RAWSXP](#) > [RawMatrix](#)
- typedef [Matrix](#)< [STRSXP](#) > [CharacterMatrix](#)
- typedef [Matrix](#)< [STRSXP](#) > [StringMatrix](#)
- typedef [Matrix](#)< [VECSXP](#) > [GenericMatrix](#)
- typedef [Matrix](#)< [VECSXP](#) > [ListMatrix](#)
- typedef [Matrix](#)< [EXPRSXP](#) > [ExpressionMatrix](#)
- typedef [WeakReference\\_Impl](#)< [PreserveStorage](#) > [WeakReference](#)

## Functions

- template<class InputIterator , class T >  
bool [\\_\\_any](#) (InputIterator first, InputIterator last, const T &value, std::input\_iterator\_tag)
- template<class RandomAccessIterator , class T >  
bool [\\_\\_any](#) (RandomAccessIterator \_\_first, RandomAccessIterator \_\_last, const T &\_\_val, std::random\_access\_iterator\_tag)
- template<class InputIterator , class T >  
bool [any](#) (InputIterator first, InputIterator last, const T &value)
- template<class InputIterator , class Predicate >  
bool [\\_\\_any\\_if](#) (InputIterator first, InputIterator last, Predicate pred, std::input\_iterator\_tag)
- template<class RandomAccessIterator , class Predicate >  
bool [\\_\\_any\\_if](#) (RandomAccessIterator \_\_first, RandomAccessIterator \_\_last, Predicate \_\_pred, std::random\_access\_iterator\_tag)
- template<class InputIterator , class Predicate >  
bool [any\\_if](#) (InputIterator first, InputIterator last, Predicate pred)
- template<> SEXP [wrap\\_extra\\_steps](#)< [Rcpp::Date](#) > (SEXP)
- template<> SEXP [wrap\\_extra\\_steps](#)< [Rcpp::Datetime](#) > (SEXP)
- template<typename T >  
SEXP [wrap\\_extra\\_steps](#) (SEXP x)
- template<> SEXP [wrap](#) (const [Date](#) &date)
- template<> SEXP [wrap](#)< [Datetime](#) > (const [Datetime](#) &date)
- [Environment new\\_env](#) (int size=29)
- [Environment new\\_env](#) (SEXP parent, int size=29)
- void [message](#) (SEXP s)

- SEXP [Rcpp\\_fast\\_eval](#) (SEXP expr, SEXP env)
- SEXP [Rcpp\\_eval](#) (SEXP expr, SEXP env)
- `template<typename T >`  
  T [as](#) (SEXP x)
- `template<> char` [as< char >](#) (SEXP x)
- `template<typename T >`  
  [traits::remove\\_const\\_and\\_reference< T >::type](#) [bare\\_as](#) (SEXP x)
- `template<> SEXP` [as](#) (SEXP x)
- `template<typename T >`  
  T [clone](#) (const T &object)
- `template<> SEXP` [wrap< Rcpp::Date >](#) (const [Rcpp::Date](#) &date)
- [Date operator+](#) (const [Date](#) &date, int offset)
- double [operator-](#) (const [Date](#) &d1, const [Date](#) &d2)
- bool [operator<](#) (const [Date](#) &d1, const [Date](#) &d2)
- bool [operator>](#) (const [Date](#) &d1, const [Date](#) &d2)
- bool [operator==](#) (const [Date](#) &d1, const [Date](#) &d2)
- bool [operator>=](#) (const [Date](#) &d1, const [Date](#) &d2)
- bool [operator<=](#) (const [Date](#) &d1, const [Date](#) &d2)
- bool [operator!=](#) (const [Date](#) &d1, const [Date](#) &d2)
- `std::ostream &` [operator<<](#) (`std::ostream &os`, const [Date](#) d)
- `template<> SEXP` [wrap< Rcpp::Datetime >](#) (const [Rcpp::Datetime](#) &dt)
- [Datetime operator+](#) (const [Datetime](#) &datetime, double offset)
- [Datetime operator+](#) (const [Datetime](#) &datetime, int offset)
- double [operator-](#) (const [Datetime](#) &d1, const [Datetime](#) &d2)
- bool [operator<](#) (const [Datetime](#) &d1, const [Datetime](#) &d2)
- bool [operator>](#) (const [Datetime](#) &d1, const [Datetime](#) &d2)
- bool [operator==](#) (const [Datetime](#) &d1, const [Datetime](#) &d2)
- bool [operator>=](#) (const [Datetime](#) &d1, const [Datetime](#) &d2)
- bool [operator<=](#) (const [Datetime](#) &d1, const [Datetime](#) &d2)
- bool [operator!=](#) (const [Datetime](#) &d1, const [Datetime](#) &d2)
- `std::ostream &` [operator<<](#) (`std::ostream &os`, const [Datetime](#) d)
- `std::ostream &` [operator<<](#) (`std::ostream &os`, const [newDatetimeVector](#) d)
- `std::ostream &` [operator<<](#) (`std::ostream &os`, const [newDateVector](#) d)
- [RCPP\\_API\\_CLASS](#) ([DottedPair\\_Impl](#))
- [RCPP\\_API\\_CLASS](#) ([Environment\\_Impl](#))
- [Environment\\_Impl](#) (SEXP x)
- [Environment\\_Impl](#) (const `std::string &name`)
- [Environment\\_Impl](#) (int pos)
- SEXP [ls](#) (bool all) const
- SEXP [get](#) (const `std::string &name`) const
- SEXP [get](#) ([Symbol](#) name) const
- SEXP [find](#) (const `std::string &name`) const
- SEXP [find](#) ([Symbol](#) name) const
- bool [exists](#) (const `std::string &name`) const
- bool [assign](#) (const `std::string &name`, SEXP x) const
- bool [assign](#) (const `std::string &name`, const [Shield](#)< SEXP > &x) const
- `template<typename WRAPPABLE >`  
  bool [assign](#) (const `std::string &name`, const WRAPPABLE &x) const
- bool [isLocked](#) () const
- bool [remove](#) (const `std::string &name`)
- void [lock](#) (bool bindings=false)

- void [lockBinding](#) (const std::string &name)
- void [unlockBinding](#) (const std::string &name)
- bool [bindingsLocked](#) (const std::string &name) const
- bool [bindingsActive](#) (const std::string &name) const
- bool [is\\_user\\_database](#) () const
- static [Environment\\_Impl global\\_env](#) ()
- static [Environment\\_Impl empty\\_env](#) ()
- static [Environment\\_Impl base\\_env](#) ()
- static [Environment\\_Impl base\\_namespace](#) ()
- static [Environment\\_Impl Rcpp\\_namespace](#) ()
- static [Environment\\_Impl namespace\\_env](#) (const std::string &package)
- [Environment\\_Impl parent](#) () const
- [Environment\\_Impl new\\_child](#) (bool hashed) const
- void [update](#) (SEXP)
- template<typename... Args>  
void [warning](#) (const char \*fmt, Args &&... args)
- template<typename... Args>  
void [NORET stop](#) (const char \*fmt, Args &&... args)
- template<typename T1 >  
void [warning](#) (const char \*fmt, const T1 &arg1)
- template<typename T1 , typename T2 >  
void [warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2)
- template<typename T1 , typename T2 , typename T3 >  
void [warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3)
- template<typename T1 , typename T2 , typename T3 , typename T4 >  
void [warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 >  
void [warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 >  
void [warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 >  
void [warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 >  
void [warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 >  
void [warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9)
- template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 , typename T10 >  
void [warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9, const T10 &arg10)
- template<typename T1 >  
void [NORET stop](#) (const char \*fmt, const T1 &arg1)
- template<typename T1 , typename T2 >  
void [NORET stop](#) (const char \*fmt, const T1 &arg1, const T2 &arg2)
- template<typename T1 , typename T2 , typename T3 >  
void [NORET stop](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3)

- `template<typename T1, typename T2, typename T3, typename T4 >`  
void [NORET stop](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4)
- `template<typename T1, typename T2, typename T3, typename T4, typename T5 >`  
void [NORET stop](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5)
- `template<typename T1, typename T2, typename T3, typename T4, typename T5, typename T6 >`  
void [NORET stop](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6)
- `template<typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7 >`  
void [NORET stop](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7)
- `template<typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7, typename T8 >`  
void [NORET stop](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8)
- `template<typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7, typename T8, typename T9 >`  
void [NORET stop](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9)
- `template<typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7, typename T8, typename T9, typename T10 >`  
void [NORET stop](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9, const T10 &arg10)
- static `std::string` [toString](#) (const int i)
- void [warning](#) (const `std::string` &message)
- void [NORET stop](#) (const `std::string` &message)
- [RCPP\\_SIMPLE\\_EXCEPTION\\_CLASS](#) (not\_a\_matrix, "Not a matrix.") [RCPP\\_SIMPLE\\_EXCEPTION\\_CLASS](#)(no\_such\_field
- No such field [RCPP\\_EXCEPTION\\_CLASS](#) (reference\_creation\_error, "Error creating object of reference class") [RCPP\\_ADVANCED\\_EXCEPTION\\_CLASS](#)(not\_compatible
- [RCPP\\_API\\_CLASS](#) ([Formula\\_Impl](#))
- [Formula\\_Impl](#) (SEXP x)
- [Formula\\_Impl](#) (const `std::string` &code)
- [RCPP\\_API\\_CLASS](#) ([Function\\_Impl](#))
- SEXP [pairlist](#) ()
- SEXP [grow](#) (SEXP head, SEXP tail)
- `template<typename T >`  
SEXP [grow](#) (const T &head, SEXP tail)
- SEXP [grow](#) (const char \*head, SEXP tail)
- [RCPP\\_API\\_CLASS](#) ([InternalFunction\\_Impl](#))
- void [checkUserInterrupt](#) ()
- `template<typename T >`  
bool [is](#) (SEXP x)
- SEXP [Rcpp\\_list2](#) (SEXP x0, SEXP x1)
- SEXP [Rcpp\\_list3](#) (SEXP x0, SEXP x1, SEXP x2)
- SEXP [Rcpp\\_list4](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3)
- SEXP [Rcpp\\_list5](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4)
- SEXP [Rcpp\\_list6](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5)
- SEXP [Rcpp\\_list7](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6)
- SEXP [Rcpp\\_lang7](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6)
- SEXP [Rcpp\\_list8](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7)
- SEXP [Rcpp\\_lang8](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7)
- SEXP [Rcpp\\_list9](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8)



- [Language\\_Impl](#) (const std::string &symbol)
- [Language\\_Impl](#) (const [Symbol](#) &symbol)
- [Language\\_Impl](#) (const [Function](#) &function)
- void [setSymbol](#) (const std::string &symbol)
- void [setSymbol](#) (const [Symbol](#) &symbol)
- void [setFunction](#) (const [Function](#) &function)
- SEXP [eval](#) () const
- SEXP [eval](#) (SEXP env) const
- SEXP [fast\\_eval](#) () const
- SEXP [fast\\_eval](#) (SEXP env) const
- template<typename RESULT\_TYPE >  
std::string [get\\_return\\_type\\_dispatch](#) ([Rcpp::traits::false\\_type](#))
- template<typename RESULT\_TYPE >  
std::string [get\\_return\\_type\\_dispatch](#) ([Rcpp::traits::true\\_type](#))
- template<typename RESULT\_TYPE >  
std::string [get\\_return\\_type](#) ()
- template<> std::string [get\\_return\\_type](#)< [void\\_type](#) > ()
- template<> std::string [get\\_return\\_type](#)< SEXP > ()
- template<> std::string [get\\_return\\_type](#)< [Rcpp::IntegerVector](#) > ()
- template<> std::string [get\\_return\\_type](#)< [Rcpp::NumericVector](#) > ()
- template<> std::string [get\\_return\\_type](#)< [Rcpp::RawVector](#) > ()
- template<> std::string [get\\_return\\_type](#)< [Rcpp::ExpressionVector](#) > ()
- template<> std::string [get\\_return\\_type](#)< [Rcpp::List](#) > ()
- template<> std::string [get\\_return\\_type](#)< [Rcpp::CharacterVector](#) > ()
- template<typename FROM , typename TO >  
void [converter](#) (const char \*from, const char \*to, TO(\*fun)(FROM), const char \*docstring=0)
- [LogicalVector shush\\_about\\_NA](#) ()
- [Argument Named](#) (const std::string &name)
- template<typename T >  
[traits::named\\_object](#)< T > [Named](#) (const std::string &name, const T &o)
- [RCPP\\_API\\_CLASS](#) ([Pairlist\\_Impl](#))
- [Pairlist\\_Impl](#) ()
- [Pairlist\\_Impl](#) (SEXP x)
- void [print](#) (SEXP s)
- void [warningcall](#) (SEXP call, const std::string &s)
- [RCPP\\_API\\_CLASS](#) ([Promise\\_Impl](#))
- SEXP [Rcpp\\_protect](#) (SEXP x)
- void [Rcpp\\_unprotect](#) (int i)
- template<int TARGET>  
SEXP [r\\_cast](#) (SEXP x)
- [RCPP\\_API\\_CLASS](#) ([Reference\\_Impl](#))
- [RCPP\\_API\\_CLASS](#) ([RObject\\_Impl](#))
- attribute\_hidden const char \* [type2name](#) (SEXP x)
- attribute\_hidden double [mktime00](#) (struct [tm](#) &[tm](#))
- attribute\_hidden struct [tm](#) \* [gmtime\\_](#) (const time\_t \*const x)
- attribute\_hidden void [Rcpp\\_precious\\_init](#) ()
- attribute\_hidden void [Rcpp\\_precious\\_teardown](#) ()
- attribute\_hidden SEXP [Rcpp\\_precious\\_preserve](#) (SEXP [object](#))
- attribute\_hidden void [Rcpp\\_precious\\_remove](#) (SEXP token)
- attribute\_hidden [Rostream](#)< true > & [Rcpp\\_cout\\_get](#) ()
- attribute\_hidden [Rostream](#)< false > & [Rcpp\\_cerr\\_get](#) ()



- [RCPP\\_API\\_CLASS](#) (S4\_Impl)
- `template<int MAX_SIZE>`  
`std::string sprintf (const char *format,...)`
- `template<bool NA, typename T >`  
`stats::D1< REALSXP, NA, T > dexp (const Rcpp::VectorBase< REALSXP, NA, T > &x, double shape, bool log=false)`
- `template<bool NA, typename T >`  
`stats::P1< REALSXP, NA, T > pexp (const Rcpp::VectorBase< REALSXP, NA, T > &x, double shape, bool lower=true, bool log=false)`
- `template<bool NA, typename T >`  
`stats::Q1< REALSXP, NA, T > qexp (const Rcpp::VectorBase< REALSXP, NA, T > &x, double shape, bool lower=true, bool log=false)`
- [NumericVector rnorm](#) (int n, double [mean](#), double [sd](#))
- [NumericVector rnorm](#) (int n, double [mean](#))
- [NumericVector rnorm](#) (int n)
- [NumericVector rbeta](#) (int n, double a, double b)
- [NumericVector rbinom](#) (int n, double nin, double pp)
- [NumericVector rcauchy](#) (int n, double location, double scale)
- [NumericVector rcauchy](#) (int n, double location)
- [NumericVector rcauchy](#) (int n)
- [NumericVector rchisq](#) (int n, double df)
- [NumericVector rexp](#) (int n, double rate)
- [NumericVector rexp](#) (int n)
- [NumericVector rf](#) (int n, double n1, double n2)
- [NumericVector rgamma](#) (int n, double a, double scale)
- [NumericVector rgamma](#) (int n, double a)
- [NumericVector rgeom](#) (int n, double p)
- [NumericVector rhyper](#) (int n, double nn1, double nn2, double kk)
- [NumericVector rlnorm](#) (int n, double meanlog, double sdlog)
- [NumericVector rlnorm](#) (int n, double meanlog)
- [NumericVector rlnorm](#) (int n)
- [NumericVector rlogis](#) (int n, double location, double scale)
- [NumericVector rlogis](#) (int n, double location)
- [NumericVector rlogis](#) (int n)
- [NumericVector rlnbinom](#) (int n, double siz, double prob)
- [NumericVector rlnbinom\\_mu](#) (int n, double siz, double mu)
- [NumericVector rnchisq](#) (int n, double df, double lambda)
- [NumericVector rnchisq](#) (int n, double df)
- [NumericVector rpois](#) (int n, double mu)
- [NumericVector rsignrank](#) (int n, double nn)
- [NumericVector rt](#) (int n, double df)
- [NumericVector runif](#) (int n, double [min](#), double [max](#))
- [NumericVector runif](#) (int n, double [min](#))
- [NumericVector runif](#) (int n)
- [NumericVector rweibull](#) (int n, double shape, double scale)
- [NumericVector rweibull](#) (int n, double shape)
- [NumericVector rwilcox](#) (int n, double mm, double nn)
- [RCPP\\_API\\_CLASS](#) ([StretchyList\\_Impl](#))
- [StretchyList\\_Impl](#) ()
- [StretchyList\\_Impl](#) (SEXP x)
- [operator SEXP](#) () const

- `template<typename T >`  
`StretchyList_Impl` & `push_back` (const T &obj)
- `template<typename T >`  
`StretchyList_Impl` & `push_front` (const T &obj)
- `template<typename T >`  
`StretchyList_Impl` & `push_back__impl` (const T &obj, traits::true\_type)
- `template<typename T >`  
`StretchyList_Impl` & `push_back__impl` (const T &obj, traits::false\_type)
- `template<typename T >`  
`StretchyList_Impl` & `push_front__impl` (const T &obj, traits::true\_type)
- `template<typename T >`  
`StretchyList_Impl` & `push_front__impl` (const T &obj, traits::false\_type)
- `template<> SEXP wrap< Rcpp::String >` (const Rcpp::String &object)
- `bool operator==` (const String::StringProxy &lhs, const String &rhs)
- `bool operator!=` (const String::StringProxy &lhs, const String &rhs)
- `bool operator==` (const String::const\_StringProxy &lhs, const String &rhs)
- `bool operator!=` (const String::const\_StringProxy &lhs, const String &rhs)
- `template<typename UnaryOperator >`  
`StringTransformer< UnaryOperator >` `make_string_transformer` (const UnaryOperator &fun)
- `template<bool NA, typename T >`  
`sugar::All< NA, T >` `all` (const Rcpp::VectorBase< LGLSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Any< NA, T >` `any` (const Rcpp::VectorBase< LGLSXP, NA, T > &t)
- `template<int RTYPE, bool NA, typename T >`  
`sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >` `clamp` (typename Rcpp::traits::storage\_type< RTYPE >::type lhs, const Rcpp::VectorBase< RTYPE, NA, T > &vec, typename Rcpp::traits::storage\_type< RTYPE >::type rhs)
- `template<bool NA, typename T >`  
`sugar::Cummax< INTSXP, NA, T >` `cummax` (const VectorBase< INTSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Cummax< REALSXP, NA, T >` `cummax` (const VectorBase< REALSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Cummin< INTSXP, NA, T >` `cummin` (const VectorBase< INTSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Cummin< REALSXP, NA, T >` `cummin` (const VectorBase< REALSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Cumprod< INTSXP, NA, T >` `cumprod` (const VectorBase< INTSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Cumprod< REALSXP, NA, T >` `cumprod` (const VectorBase< REALSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Cumprod< CPLXSXP, NA, T >` `cumprod` (const VectorBase< CPLXSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Cumsum< INTSXP, NA, T >` `cumsum` (const VectorBase< INTSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Cumsum< REALSXP, NA, T >` `cumsum` (const VectorBase< REALSXP, NA, T > &t)
- `template<bool LHS_NA, typename LHS_T >`  
`sugar::Diff< INTSXP, LHS_NA, LHS_T >` `diff` (const VectorBase< INTSXP, LHS\_NA, LHS\_T > &lhs)
- `template<bool LHS_NA, typename LHS_T >`  
`sugar::Diff< REALSXP, LHS_NA, LHS_T >` `diff` (const VectorBase< REALSXP, LHS\_NA, LHS\_T > &lhs)
- `template<int RTYPE, bool NA, typename T >`  
`LogicalVector` `duplicated` (const VectorBase< RTYPE, NA, T > &x)
- `template<int RTYPE, bool NA, typename T >`  
`sugar::Head< RTYPE, NA, T >` `head` (const VectorBase< RTYPE, NA, T > &t, R\_xlen\_t n)

- `template<int RTYPE, bool COND_NA, typename COND_T, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > ifelse` (const `Rcpp::VectorBase< LGLSXP, COND_NA, COND_T >` &cond, const `Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T >` &lhs, const `Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T >` &rhs)
- `template<int RTYPE, bool COND_NA, typename COND_T, bool RHS_NA, typename RHS_T >`  
`sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T > ifelse` (const `Rcpp::VectorBase< LGLSXP, COND_NA, COND_T >` &cond, typename `traits::storage_type< RTYPE >::type` lhs, const `Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T >` &rhs)
- `template<int RTYPE, bool COND_NA, typename COND_T, bool RHS_NA, typename RHS_T >`  
`sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T > ifelse` (const `Rcpp::VectorBase< LGLSXP, COND_NA, COND_T >` &cond, const `Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T >` &lhs, typename `traits::storage_type< RTYPE >::type` rhs)
- `template<bool COND_NA, typename COND_T >`  
`sugar::IfElse_Primitive_Primitive< REALSXP, COND_NA, COND_T > ifelse` (const `Rcpp::VectorBase< LGLSXP, COND_NA, COND_T >` &cond, double lhs, double rhs)
- `template<bool COND_NA, typename COND_T >`  
`sugar::IfElse_Primitive_Primitive< INTSXP, COND_NA, COND_T > ifelse` (const `Rcpp::VectorBase< LGLSXP, COND_NA, COND_T >` &cond, int lhs, int rhs)
- `template<bool COND_NA, typename COND_T >`  
`sugar::IfElse_Primitive_Primitive< CPLXSXP, COND_NA, COND_T > ifelse` (const `Rcpp::VectorBase< LGLSXP, COND_NA, COND_T >` &cond, Rcomplex lhs, Rcomplex rhs)
- `template<bool COND_NA, typename COND_T >`  
`sugar::IfElse_Primitive_Primitive< LGLSXP, COND_NA, COND_T > ifelse` (const `Rcpp::VectorBase< LGLSXP, COND_NA, COND_T >` &cond, bool lhs, bool rhs)
- `template<int RTYPE, bool NA, typename T >`  
`sugar::IsFinite< RTYPE, NA, T > is_finite` (const `Rcpp::VectorBase< RTYPE, NA, T >` &t)
- `template<int RTYPE, bool NA, typename T >`  
`sugar::IsInfinite< RTYPE, NA, T > is_infinite` (const `Rcpp::VectorBase< RTYPE, NA, T >` &t)
- `template<int RTYPE, bool NA, typename T >`  
`sugar::IsNa< RTYPE, NA, T > is_na` (const `Rcpp::VectorBase< RTYPE, NA, T >` &t)
- `sugar::IsNa_Vector_is_na< oldDatetimeVector > is_na` (const `oldDatetimeVector` &x)
- `sugar::IsNa_Vector_is_na< oldDateVector > is_na` (const `oldDateVector` &x)
- `sugar::IsNa_Vector_is_na< NumericVector > is_na` (`newDatetimeVector` &x)
- `sugar::IsNa_Vector_is_na< NumericVector > is_na` (`newDateVector` &x)
- `template<int RTYPE, bool NA, typename T >`  
`sugar::IsNaN< RTYPE, NA, T > is_nan` (const `Rcpp::VectorBase< RTYPE, NA, T >` &t)
- `template<int RTYPE, bool NA, typename T, typename Function >`  
`sugar::Lapply< RTYPE, NA, T, Function > lapply` (const `Rcpp::VectorBase< RTYPE, NA, T >` &t, `Function` fun)
- `template<int RTYPE, bool NA_1, typename T_1, bool NA_2, typename T_2, typename Function >`  
`sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function > mapply` (const `Rcpp::VectorBase< RTYPE, NA_1, T_1 >` &t1, const `Rcpp::VectorBase< RTYPE, NA_2, T_2 >` &t2, `Function` fun)
- `template<int RTYPE, bool NA_1, typename T_1, typename Function >`  
`sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, double, Function > mapply` (const `Rcpp::VectorBase< RTYPE, NA_1, T_1 >` &t1, double t2, `Function` fun)
- `template<int RTYPE, bool NA_2, typename T_2, typename Function >`  
`sugar::Mapply_2_Primitive_Vector< RTYPE, double, NA_2, T_2, Function > mapply` (double t1, const `Rcpp::VectorBase< RTYPE, NA_2, T_2 >` &t2, `Function` fun)
- `template<int RTYPE_1, bool NA_1, typename T_1, int RTYPE_2, bool NA_2, typename T_2, int RTYPE_3, bool NA_3, typename T_3, typename Function >`  
`sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function > mapply` (const `Rcpp::VectorBase< RTYPE_1, NA_1, T_1 >` &t1, const `Rcpp::VectorBase< RTYPE_2, NA_2, T_2 >` &t2, const `Rcpp::VectorBase< RTYPE_3, NA_3, T_3 >` &t3, `Function` fun)

- `template<int RTYPE, bool NA, typename T, bool RHS_NA, typename RHS_T >`  
[IntegerVector match](#) (const [VectorBase](#)< RTYPE, NA, T > &x, const [VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &table\_)
- `template<int RTYPE, bool NA, typename T >`  
[sugar::Max](#)< RTYPE, NA, T > [max](#) (const [VectorBase](#)< RTYPE, NA, T > &x)
- `template<bool NA, typename T >`  
[sugar::Mean](#)< REALSXP, NA, T > [mean](#) (const [VectorBase](#)< REALSXP, NA, T > &t)
- `template<bool NA, typename T >`  
[sugar::Mean](#)< INTSXP, NA, T > [mean](#) (const [VectorBase](#)< INTSXP, NA, T > &t)
- `template<bool NA, typename T >`  
[sugar::Mean](#)< CPLXSXP, NA, T > [mean](#) (const [VectorBase](#)< CPLXSXP, NA, T > &t)
- `template<bool NA, typename T >`  
[sugar::Mean](#)< LGLSXP, NA, T > [mean](#) (const [VectorBase](#)< LGLSXP, NA, T > &t)
- `template<int RTYPE, bool NA, typename T >`  
[sugar::median\\_detail::result](#)< RTYPE >::type [median](#) (const [Rcpp::VectorBase](#)< RTYPE, NA, T > &x, bool na←\_rm=false)
- `template<int RTYPE, bool NA, typename T >`  
[sugar::Min](#)< RTYPE, NA, T > [min](#) (const [VectorBase](#)< RTYPE, NA, T > &x)
- `template<int RTYPE, bool NA, typename T >`  
[Vector](#)< RTYPE > [na\\_omit](#) (const [VectorBase](#)< RTYPE, NA, T > &t)
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
[sugar::Pmax\\_Vector\\_Vector](#)< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > [pmax](#) (const [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, const [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T >`  
[sugar::Pmax\\_Vector\\_Primitive](#)< RTYPE, LHS\_NA, LHS\_T > [pmax](#) (const [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, typename [Rcpp::traits::storage\\_type](#)< RTYPE >::type rhs)
- `template<int RTYPE, bool RHS_NA, typename RHS_T >`  
[sugar::Pmax\\_Vector\\_Primitive](#)< RTYPE, RHS\_NA, RHS\_T > [pmax](#) (typename [Rcpp::traits::storage\\_type](#)< RTYPE >::type lhs, const [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
[sugar::Pmin\\_Vector\\_Vector](#)< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > [pmin](#) (const [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, const [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T >`  
[sugar::Pmin\\_Vector\\_Primitive](#)< RTYPE, LHS\_NA, LHS\_T > [pmin](#) (const [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, typename [Rcpp::traits::storage\\_type](#)< RTYPE >::type rhs)
- `template<int RTYPE, bool RHS_NA, typename RHS_T >`  
[sugar::Pmin\\_Vector\\_Primitive](#)< RTYPE, RHS\_NA, RHS\_T > [pmin](#) (typename [Rcpp::traits::storage\\_type](#)< RTYPE >::type lhs, const [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)
- `template<int RTYPE, bool NA, typename T, typename EXPONENT_TYPE >`  
[sugar::Pow](#)< RTYPE, NA, T, EXPONENT\_TYPE > [pow](#) (const [VectorBase](#)< RTYPE, NA, T > &t, EXPONENT←\_TYPE exponent)
- `template<int RTYPE, bool NA, typename T >`  
[sugar::Range](#)< RTYPE, NA, T > [range](#) (const [VectorBase](#)< RTYPE, NA, T > &x)
- `template<int RTYPE, bool NA, typename T >`  
[sugar::Rep](#)< RTYPE, NA, T > [rep](#) (const [VectorBase](#)< RTYPE, NA, T > &t, R\_xlen\_t n)
- [sugar::Rep\\_Single](#)< double > [rep](#) (const double &x, R\_xlen\_t n)
- [sugar::Rep\\_Single](#)< int > [rep](#) (const int &x, R\_xlen\_t n)
- [sugar::Rep\\_Single](#)< Rbyte > [rep](#) (const Rbyte &x, R\_xlen\_t n)
- [sugar::Rep\\_Single](#)< Rcomplex > [rep](#) (const Rcomplex &x, R\_xlen\_t n)
- [sugar::Rep\\_Single](#)< bool > [rep](#) (const bool &x, R\_xlen\_t n)
- `template<int RTYPE, bool NA, typename T >`  
[sugar::Rep\\_each](#)< RTYPE, NA, T > [rep\\_each](#) (const [VectorBase](#)< RTYPE, NA, T > &t, R\_xlen\_t times)

- `template<int RTYPE, bool NA, typename T >`  
[sugar::Rep\\_len](#)< RTYPE, NA, T > [rep\\_len](#) (const [VectorBase](#)< RTYPE, NA, T > &t, R\_xlen\_t len)
- `template<int RTYPE, bool NA, typename T >`  
[sugar::Rev](#)< RTYPE, NA, T > [rev](#) (const [VectorBase](#)< RTYPE, NA, T > &t)
- `template<int RTYPE, bool NA, typename T >`  
[sugar::detail::RowSumsReturn](#)< RTYPE >::type [rowSums](#) (const [MatrixBase](#)< RTYPE, NA, T > &x, bool na←\_rm=false)
- `template<int RTYPE, bool NA, typename T >`  
[sugar::detail::ColSumsReturn](#)< RTYPE >::type [colSums](#) (const [MatrixBase](#)< RTYPE, NA, T > &x, bool na←rm=false)
- `template<int RTYPE, bool NA, typename T >`  
[sugar::detail::RowMeansReturn](#)< RTYPE >::type [rowMeans](#) (const [MatrixBase](#)< RTYPE, NA, T > &x, bool na←\_rm=false)
- `template<int RTYPE, bool NA, typename T >`  
[sugar::detail::ColMeansReturn](#)< RTYPE >::type [colMeans](#) (const [MatrixBase](#)< RTYPE, NA, T > &x, bool na←\_rm=false)
- [Vector](#)< INTSXP > [sample](#) (int n, int size, bool replace=false, [sugar::probs\\_t](#) probs=R\_NilValue, bool one←based=true)
- `template<int RTYPE>`  
[Vector](#)< RTYPE > [sample](#) (const [Vector](#)< RTYPE > &x, int size, bool replace=false, [sugar::probs\\_t](#) probs=R←\_NilValue)
- `template<int RTYPE, bool NA, typename T, typename Function >`  
[sugar::Sapply](#)< RTYPE, NA, T, Function, traits::same\_type< typename ::Rcpp::sugar::sapply\_application\_result\_of< Function, T >::type, typename Rcpp::traits::storage\_type< traits::r\_sexptype\_traits< typename ::Rcpp::sugar::sapply\_application\_r Function, T >::type >::type >::value > [sapply](#) (const [Rcpp::VectorBase](#)< RTYPE, NA, T > &t, Function fun)
- `template<bool NA, typename T >`  
[sugar::Sd](#)< REALSXP, NA, T > [sd](#) (const [VectorBase](#)< REALSXP, NA, T > &t)
- `template<int RTYPE, bool NA, typename T >`  
[IntegerVector self\\_match](#) (const [VectorBase](#)< RTYPE, NA, T > &x)
- `template<int RTYPE, bool NA, typename T >`  
[sugar::SeqLen seq\\_along](#) (const [Rcpp::VectorBase](#)< RTYPE, NA, T > &t)
- [sugar::SeqLen seq\\_len](#) (const size\_t &n)
- [Range seq](#) (R\_xlen\_t start, R\_xlen\_t end)
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
[Vector](#)< RTYPE > [setdiff](#) (const [VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, const [VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
bool [setequal](#) (const [VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, const [VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
[Vector](#)< RTYPE > [intersect](#) (const [VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, const [VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
[Vector](#)< RTYPE > [union\\_](#) (const [VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, const [VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)
- `template<bool NA, typename T >`  
[sugar::Sign](#)< INTSXP, NA, T > [sign](#) (const [VectorBase](#)< INTSXP, NA, T > &t)
- `template<bool NA, typename T >`  
[sugar::Sign](#)< REALSXP, NA, T > [sign](#) (const [VectorBase](#)< REALSXP, NA, T > &t)
- `template<bool NA, typename T >`  
[String collapse](#) (const [VectorBase](#)< STRSXP, NA, T > &vec)
- [Vector](#)< STRSXP > [trimws](#) (const [Vector](#)< STRSXP > &x, const char \*which="both")

- [Matrix](#)< STRSXP > [trimws](#) (const [Matrix](#)< STRSXP > &x, const char \*which="both")
- [String](#) [trimws](#) (const [String](#) &str, const char \*which="both")
- template<bool NA, typename T >  
[sugar::Sum](#)< INTSXP, NA, T > [sum](#) (const [VectorBase](#)< INTSXP, NA, T > &t)
- template<bool NA, typename T >  
[sugar::Sum](#)< REALSXP, NA, T > [sum](#) (const [VectorBase](#)< REALSXP, NA, T > &t)
- template<bool NA, typename T >  
[sugar::Sum](#)< LGLSXP, NA, T > [sum](#) (const [VectorBase](#)< LGLSXP, NA, T > &t)
- template<int RTYPE, bool NA, typename T >  
[IntegerVector](#) [table](#) (const [VectorBase](#)< RTYPE, NA, T > &x)
- template<int RTYPE, bool NA, typename T >  
[sugar::Tail](#)< RTYPE, NA, T > [tail](#) (const [VectorBase](#)< RTYPE, NA, T > &t, R\_xlen\_t n)
- template<int RTYPE, bool NA, typename T >  
[Vector](#)< RTYPE > [unique](#) (const [VectorBase](#)< RTYPE, NA, T > &t)
- template<int RTYPE, bool NA, typename T >  
[Vector](#)< RTYPE > [sort\\_unique](#) (const [VectorBase](#)< RTYPE, NA, T > &t, bool decreasing=false)
- template<int RTYPE, bool NA, typename T, bool RHS\_NA, typename RHS\_T >  
[LogicalVector](#) [in](#) (const [VectorBase](#)< RTYPE, NA, T > &x, const [VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &table)
- template<bool NA, typename T >  
[sugar::Var](#)< REALSXP, NA, T > [var](#) (const [VectorBase](#)< REALSXP, NA, T > &t)
- template<bool NA, typename T >  
[sugar::Var](#)< INTSXP, NA, T > [var](#) (const [VectorBase](#)< INTSXP, NA, T > &t)
- template<bool NA, typename T >  
[sugar::Var](#)< LGLSXP, NA, T > [var](#) (const [VectorBase](#)< LGLSXP, NA, T > &t)
- template<bool NA, typename T >  
[sugar::Var](#)< CPLXSXP, NA, T > [var](#) (const [VectorBase](#)< CPLXSXP, NA, T > &t)
- template<int RTYPE, bool NA, typename T >  
R\_xlen\_t [which\\_max](#) (const [VectorBase](#)< RTYPE, NA, T > &t)
- template<int RTYPE, bool NA, typename T >  
R\_xlen\_t [which\\_min](#) (const [VectorBase](#)< RTYPE, NA, T > &t)
- template<bool NA, typename T >  
bool [is\\_true](#) (const [Rcpp::sugar::SingleLogicalResult](#)< NA, T > &x)
- template<bool NA, typename T >  
bool [is\\_false](#) (const [Rcpp::sugar::SingleLogicalResult](#)< NA, T > &x)
- template<bool NA, typename T >  
bool [is\\_na](#) (const [Rcpp::sugar::SingleLogicalResult](#)< NA, T > &x)
- template<int RTYPE, bool NA, typename T >  
[Rcpp::Vector](#)< RTYPE > [as\\_vector](#) (const [MatrixBase](#)< RTYPE, NA, T > &t)
- template<int RTYPE, bool LHS\_NA, typename LHS\_T >  
[sugar::Col](#)< RTYPE, LHS\_NA, LHS\_T > [col](#) (const [Rcpp::MatrixBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs)
- template<typename T >  
[sugar::diag\\_result\\_type\\_trait](#)< T >::type [diag](#) (const T &t)
- template<int RTYPE, bool NA, typename T >  
[sugar::LowerTri](#)< RTYPE, NA, T > [lower\\_tri](#) (const [Rcpp::MatrixBase](#)< RTYPE, NA, T > &lhs, bool [diag](#)=false)
- template<int RTYPE, bool LHS\_NA, typename LHS\_T, bool RHS\_NA, typename RHS\_T, typename Function >  
[sugar::Outer](#)< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function > [outer](#) (const [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, const [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs, [Function](#) fun)
- template<int RTYPE, bool LHS\_NA, typename LHS\_T >  
[sugar::Row](#)< RTYPE, LHS\_NA, LHS\_T > [row](#) (const [Rcpp::MatrixBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs)
- template<int RTYPE, bool NA, typename T >  
[sugar::UpperTri](#)< RTYPE, NA, T > [upper\\_tri](#) (const [Rcpp::MatrixBase](#)< RTYPE, NA, T > &lhs, bool [diag](#)=false)



- `template<int RTYPE, bool NA, typename VECTOR >`  
`sugar::Nona< RTYPE, NA, VECTOR > noNA (const Rcpp::VectorBase< RTYPE, NA, VECTOR > &vec)`
- `sugar::NonaPrimitive< double > noNA (double x)`
- `sugar::NonaPrimitive< int > noNA (int x)`
- `template<int RTYPE, bool NA, typename T , typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Divides_Vector_Primitive< RTYPE, NA, T > >::type operator/ (const VectorBase< RTYPE, NA, T > &lhs, const U &rhs)`
- `template<int RTYPE, bool NA, typename T , typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Divides_Primitive_Vector< RTYPE, NA, T > >::type operator/ (const U &lhs, const VectorBase< RTYPE, NA, T > &rhs)`
- `template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >`  
`sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > operator/ (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)`
- `template<int RTYPE, bool NA, typename T , typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Minus_Vector_Primitive< RTYPE, NA, T > >::type operator- (const VectorBase< RTYPE, NA, T > &lhs, const U &rhs)`
- `template<int RTYPE, bool NA, typename T , typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Minus_Primitive_Vector< RTYPE, NA, T > >::type operator- (const U &lhs, const VectorBase< RTYPE, NA, T > &rhs)`
- `template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >`  
`sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > operator- (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)`
- `template<int RTYPE, bool NA, typename T , typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, typename sugar::Plus_Vector_Primitive< RTYPE, NA, T > >::type operator+ (const VectorBase< RTYPE, NA, T > &lhs, const U &rhs)`
- `template<int RTYPE, bool NA, typename T , typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, typename sugar::Plus_Vector_Primitive< RTYPE, NA, T > >::type operator+ (const U &rhs, const VectorBase< RTYPE, NA, T > &lhs)`
- `template<int RTYPE, bool NA, typename T , typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T > >::type operator+ (const VectorBase< RTYPE, NA, T > &lhs, const typename sugar::NonaPrimitive< U > &rhs)`
- `template<int RTYPE, bool NA, typename T , typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T > >::type operator+ (const typename sugar::NonaPrimitive< U > &rhs, const VectorBase< RTYPE, NA, T > &lhs)`
- `template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >`  
`sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > operator+ (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)`
- `template<int RTYPE, bool NA, typename T , typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Times_Vector_Primitive< RTYPE, NA, T > >::type operator* (const VectorBase< RTYPE, NA, T > &lhs, const U &rhs)`
- `template<int RTYPE, bool NA, typename T , typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Times_Vector_Primitive< RTYPE, NA, T > >::type operator* (const U &rhs, const VectorBase< RTYPE, NA, T > &lhs)`

- `template<int RTYPE, bool NA, typename T, typename U >`  
[traits::enable\\_if< traits::is\\_convertible< typename traits::remove\\_const\\_and\\_reference< U >::type, typename traits::storage\\_type< RTYPE >::type >::value, sugar::Times\\_Vector\\_Primitive\\_nona< RTYPE, NA, T > >::type](#)  
[operator\\*](#) (const [VectorBase< RTYPE, NA, T >](#) &lhs, const typename [sugar::NonaPrimitive< U >](#) &rhs)
- `template<int RTYPE, bool NA, typename T, typename U >`  
[traits::enable\\_if< traits::is\\_convertible< typename traits::remove\\_const\\_and\\_reference< U >::type, typename traits::storage\\_type< RTYPE >::type >::value, sugar::Times\\_Vector\\_Primitive\\_nona< RTYPE, NA, T > >::type](#)  
[operator\\*](#) (const typename [sugar::NonaPrimitive< U >](#) &rhs, const [VectorBase< RTYPE, NA, T >](#) &lhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
[sugar::Times\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#) [operator\\*](#) (const [VectorBase< RTYPE, LHS\\_NA, LHS\\_T >](#) &lhs, const [VectorBase< RTYPE, RHS\\_NA, RHS\\_T >](#) &rhs)
- [RCPP\\_API\\_CLASS](#) ([Symbol\\_Impl](#))
- `SEXP` [unwindProtect](#) (`SEXP` (\*callback)(void \*data), void \*data)
- `template<typename T, typename Function >`  
[List](#) [lapply](#) (const [ListOf< T >](#) &t, [Function](#) fun)
- `template<typename T, typename Function >`  
[T](#) [sapply](#) (const [ListOf< T >](#) &t, [Function](#) fun)
- `internal::DimNameProxy` [rownames](#) (`SEXP` x)
- `internal::DimNameProxy` [colnames](#) (`SEXP` x)
- `template<template< class > class StoragePolicy >`  
[std::ostream](#) & [operator<<](#) ([std::ostream](#) &s, const [Matrix< REALSXP, StoragePolicy >](#) &rhs)
- `template<template< class > class StoragePolicy >`  
[std::ostream](#) & [operator<<](#) ([std::ostream](#) &s, const [Matrix< INTSXP, StoragePolicy >](#) &rhs)
- `template<template< class > class StoragePolicy >`  
[std::ostream](#) & [operator<<](#) ([std::ostream](#) &s, const [Matrix< STRSXP, StoragePolicy >](#) &rhs)
- `template<int RTYPE, template< class > class StoragePolicy >`  
[std::ostream](#) & [operator<<](#) ([std::ostream](#) &s, const [Matrix< RTYPE, StoragePolicy >](#) &rhs)
- `template<int RTYPE, template< class > class StoragePolicy >`  
[Matrix< RTYPE, StoragePolicy >](#) [transpose\\_impl](#) (const [Matrix< RTYPE, StoragePolicy >](#) &x)
- `template<template< class > class StoragePolicy >`  
[Matrix< REALSXP, StoragePolicy >](#) [transpose](#) (const [Matrix< REALSXP, StoragePolicy >](#) &x)
- `template<template< class > class StoragePolicy >`  
[Matrix< INTSXP, StoragePolicy >](#) [transpose](#) (const [Matrix< INTSXP, StoragePolicy >](#) &x)
- `template<template< class > class StoragePolicy >`  
[Matrix< STRSXP, StoragePolicy >](#) [transpose](#) (const [Matrix< STRSXP, StoragePolicy >](#) &x)
- `no_init_vector` [no\\_init](#) (`R_xlen_t` size)
- `no_init_matrix` [no\\_init](#) (`int` nr, `int` nc)
- `template<int RTYPE, template< class > class StoragePolicy >`  
[std::ostream](#) & [operator<<](#) ([std::ostream](#) &s, const [Vector< RTYPE, StoragePolicy >](#) &rhs)
- `template<template< class > class StoragePolicy >`  
[std::ostream](#) & [operator<<](#) ([std::ostream](#) &s, const [Vector< STRSXP, StoragePolicy >](#) &rhs)
- [RCPP\\_API\\_CLASS](#) ([WeakReference\\_Impl](#))
- `template<typename T >`  
void [standard\\_delete\\_finalizer](#) (`T` \*obj)
- `template<typename T, void Finalizer >`  
void [finalizer\\_wrapper](#) (`SEXP` p)
- `SEXP` [Rcpp\\_PreserveObject](#) (`SEXP` x)
- void [Rcpp\\_ReleaseObject](#) (`SEXP` x)
- `SEXP` [Rcpp\\_ReplaceObject](#) (`SEXP` x, `SEXP` y)
- `SEXP` [Rcpp\\_PreciousPreserve](#) (`SEXP` object)
- void [Rcpp\\_PreciousRelease](#) (`SEXP` token)
- static `int` [tzparse](#) (const char \*name, struct [state](#) \*sp, `int` lastditch)



- static int [typesequiv](#) (const struct [state](#) \*sp, int a, int b)
- static const char \* [getsecs](#) (const char \*strp, int\_fast32\_t \*secsp)
- static const char \* [getnum](#) (const char \*strp, int \*const nump, const int [min](#), const int [max](#))
- static const char \* [getrule](#) (const char \*strp, struct [rule](#) \*const rulep)
- static int\_fast32\_t [transtime](#) (int year, const struct [rule](#) \*rulep, int\_fast32\_t offset)
- static struct [tm](#) \* [timesub](#) (const time\_t \*timep, int\_fast32\_t offset, const struct [state](#) \*sp, struct [tm](#) \*tmp)
- static int [leaps\\_thru\\_end\\_of](#) (const int y)
- static int [increment\\_overflow](#) (int \*const ip, int j)
- static int [increment\\_overflow\\_time](#) (time\_t \*tp, int\_fast32\_t j)
- static int\_fast32\_t [detzcode](#) (const char \*const codep)
- static int\_fast64\_t [detzcode64](#) (const char \*const codep)
- static int [differ\\_by\\_repeat](#) (const time\_t t1, const time\_t t0)
- static const char \* [getzname](#) (const char \*strp)
- static const char \* [getqzname](#) (const char \*strp, const int delim)
- static const char \* [getoffset](#) (const char \*strp, int\_fast32\_t \*const offsetp)
- static int [tzload](#) (const char \*name, struct [state](#) \*const sp, const int doextend)
- static void [gmtload](#) (struct [state](#) \*const sp)
- static struct [tm](#) \* [gmtsub](#) (const time\_t \*const timep, const int\_fast32\_t offset, struct [tm](#) \*const tmp)

## Variables

- public [DottedPairProxyPolicy](#)< [DottedPair\\_Impl](#)< [StoragePolicy](#) > >
- public public [DottedPairImpl](#)< [DottedPair\\_Impl](#)< [StoragePolicy](#) > >
- public [BindingPolicy](#)< [Environment\\_Impl](#)< [StoragePolicy](#) > >
- public [DottedPairProxyPolicy](#)< [Formula\\_Impl](#)< [StoragePolicy](#) > >
- public public [DottedPairImpl](#)< [Formula\\_Impl](#)< [StoragePolicy](#) > >
- static [Rostream](#)< true > [Rcout](#) = [Rcpp\\_cout\\_get](#)()
- static [Rostream](#)< false > [Rcerr](#) = [Rcpp\\_cerr\\_get](#)()
- public [DottedPairProxyPolicy](#)< [Language\\_Impl](#)< [StoragePolicy](#) > >
- public public [DottedPairImpl](#)< [Language\\_Impl](#)< [StoragePolicy](#) > >
- static [Na\\_Proxy](#) NA
- static [internal::NamedPlaceholder](#) \_
- public [DottedPairProxyPolicy](#)< [Pairlist\\_Impl](#)< [StoragePolicy](#) > >
- public public [DottedPairImpl](#)< [Pairlist\\_Impl](#)< [StoragePolicy](#) > >
- public [FieldProxyPolicy](#)< [Reference\\_Impl](#)< [StoragePolicy](#) > >
- public [DottedPairProxyPolicy](#)< [StretchyList\\_Impl](#)< [StoragePolicy](#) > >
- static SEXP [Rcpp\\_precious](#) = R\_NilValue
- static time\_t const [time\\_t\\_min](#) = [MINVAL](#)(time\_t, [TYPE\\_BIT](#)(time\_t))
- static time\_t const [time\\_t\\_max](#) = [MAXVAL](#)(time\_t, [TYPE\\_BIT](#)(time\_t))
- static const char [gmt](#) [] = "GMT"
- static const int [mon\\_lengths](#) [2][[MONSPERYEAR](#)]
- static const int [year\\_lengths](#) [2]
- static int [gmt\\_is\\_set](#)
- static struct [state](#) [gmtmem](#)
- static struct [tm](#) [tm](#)
- static [Module](#) \* [current\\_scope](#)

### 5.2.1 Detailed Description

[Rcpp](#) API.

## 5.2.2 Typedef Documentation

### 5.2.2.1 CharacterMatrix

```
typedef Matrix<STRSXP> Rcpp::CharacterMatrix
```

Definition at line 46 of file instantiation.h.

### 5.2.2.2 CharacterVector

```
typedef Vector<STRSXP> Rcpp::CharacterVector
```

Definition at line 34 of file instantiation.h.

### 5.2.2.3 ComplexMatrix

```
typedef Matrix<CPLXSXP> Rcpp::ComplexMatrix
```

Definition at line 40 of file instantiation.h.

### 5.2.2.4 ComplexVector

```
typedef Vector<CPLXSXP> Rcpp::ComplexVector
```

Definition at line 27 of file instantiation.h.

### 5.2.2.5 const\_Proxy

```
typedef DottedPairProxyPolicy< StretchyList_Impl >::const_DottedPairProxy Rcpp::const_Proxy
```

Definition at line 38 of file Language.h.

### 5.2.2.6 DataFrame

```
typedef DataFrame_Impl<PreserveStorage> Rcpp::DataFrame
```

Definition at line 190 of file DataFrame.h.

### 5.2.2.7 DatetimeVector

```
typedef oldDatetimeVector Rcpp::DatetimeVector
```

Definition at line 45 of file date\_datetime.h.

### 5.2.2.8 DateVector

```
typedef oldDateVector Rcpp::DateVector
```

Definition at line 44 of file date\_datetime.h.

### 5.2.2.9 DottedPair

```
typedef DottedPair_Impl<PreserveStorage> Rcpp::DottedPair
```

Definition at line 45 of file DottedPair.h.

### 5.2.2.10 DoubleVector

```
typedef Vector<REALSXP> Rcpp::DoubleVector
```

Definition at line 31 of file instantiation.h.

### 5.2.2.11 Environment

```
typedef Environment_Impl<PreserveStorage> Rcpp::Environment
```

Definition at line 403 of file Environment.h.

### 5.2.2.12 ExpressionMatrix

```
typedef Matrix<EXPRXP> Rcpp::ExpressionMatrix
```

Definition at line 50 of file instantiation.h.

### 5.2.2.13 ExpressionVector

```
typedef Vector<EXPRXP> Rcpp::ExpressionVector
```

Definition at line 38 of file instantiation.h.

### 5.2.2.14 Formula

```
typedef Formula_Impl<PreserveStorage> Rcpp::Formula
```

Definition at line 77 of file Formula.h.

### 5.2.2.15 Function

```
typedef Function_Impl<PreserveStorage> Rcpp::Function
```

#### Examples

[SugarPerformance/Timer.h](#).

Definition at line 124 of file Function.h.

### 5.2.2.16 GenericMatrix

```
typedef Matrix<VECSXP> Rcpp::GenericMatrix
```

Definition at line 48 of file instantiation.h.

### 5.2.2.17 GenericVector

```
typedef Vector<VECSXP> Rcpp::GenericVector
```

Definition at line 36 of file instantiation.h.

### 5.2.2.18 IntegerMatrix

```
typedef Matrix<INTSXP> Rcpp::IntegerMatrix
```

Definition at line 41 of file instantiation.h.

### 5.2.2.19 IntegerVector

```
typedef Vector<INTSXP> Rcpp::IntegerVector
```

Definition at line 28 of file instantiation.h.

### 5.2.2.20 InternalFunction

```
typedef InternalFunction_Impl<PreserveStorage> Rcpp::InternalFunction
```

Definition at line 64 of file InternalFunction.h.

### 5.2.2.21 Language

```
typedef Language_Impl<PreserveStorage> Rcpp::Language
```

Definition at line 160 of file Language.h.

### 5.2.2.22 List

```
typedef Vector<VECSXP> Rcpp::List
```

Definition at line 37 of file instantiation.h.

### 5.2.2.23 ListMatrix

```
typedef Matrix<VECSXP> Rcpp::ListMatrix
```

Definition at line 49 of file instantiation.h.

### 5.2.2.24 LogicalMatrix

```
typedef Matrix<LGLSXP> Rcpp::LogicalMatrix
```

Definition at line 42 of file instantiation.h.

### 5.2.2.25 LogicalVector

```
typedef Vector<LGLSXP> Rcpp::LogicalVector
```

Definition at line 29 of file instantiation.h.

### 5.2.2.26 nanotime\_t

```
typedef uint64_t Rcpp::nanotime_t
```

Definition at line 47 of file Timer.h.

### 5.2.2.27 NumericMatrix

```
typedef Matrix<REALSXP> Rcpp::NumericMatrix
```

Definition at line 43 of file instantiation.h.

### 5.2.2.28 NumericVector

```
typedef Vector<REALSXP> Rcpp::NumericVector
```

Definition at line 30 of file instantiation.h.

### 5.2.2.29 Pairlist

```
typedef Pairlist_Impl<PreserveStorage> Rcpp::Pairlist
```

Definition at line 53 of file Pairlist.h.

### 5.2.2.30 Promise

```
typedef Promise_Impl<PreserveStorage> Rcpp::Promise
```

Definition at line 78 of file Promise.h.

### 5.2.2.31 RawMatrix

```
typedef Matrix<RAWSXP> Rcpp::RawMatrix
```

Definition at line 44 of file instantiation.h.

### 5.2.2.32 RawVector

```
typedef Vector<RAWSXP> Rcpp::RawVector
```

Definition at line 32 of file instantiation.h.

### 5.2.2.33 Reference

```
typedef Reference_Impl<PreserveStorage> Rcpp::Reference
```

Definition at line 71 of file Reference.h.

### 5.2.2.34 RObject

```
typedef RObject_Impl<PreserveStorage> Rcpp::RObject
```

Definition at line 58 of file RObject.h.

### 5.2.2.35 S4

```
typedef S4_Impl<PreserveStorage> Rcpp::S4
```

Definition at line 77 of file S4.h.

### 5.2.2.36 StretchyList

```
typedef StretchyList_Impl<PreserveStorage> Rcpp::StretchyList [private]
```

Definition at line 83 of file StretchyList.h.

### 5.2.2.37 StringMatrix

```
typedef Matrix<STRSXP> Rcpp::StringMatrix
```

Definition at line 47 of file instantiation.h.

### 5.2.2.38 StringVector

```
typedef Vector<STRSXP> Rcpp::StringVector
```

Definition at line 35 of file instantiation.h.

### 5.2.2.39 Symbol

```
typedef Symbol_Impl<NoProtectStorage> Rcpp::Symbol
```

Definition at line 84 of file Symbol.h.

### 5.2.2.40 ValidConstructor

```
typedef bool(* Rcpp::ValidConstructor) (SEXP *, int)
```

Definition at line 139 of file Module.h.



### 5.2.2.41 ValidMethod

```
typedef bool(* Rcpp::ValidMethod) (SEXP *, int)
```

Definition at line 140 of file Module.h.

### 5.2.2.42 WeakReference

```
typedef WeakReference_Impl<PreserveStorage> Rcpp::WeakReference
```

Definition at line 64 of file WeakReference.h.

## 5.2.3 Function Documentation

### 5.2.3.1 \_\_any() [1/2]

```
template<class InputIterator , class T >
bool Rcpp::__any (
    InputIterator first,
    InputIterator last,
    const T & value,
    std::input_iterator_tag ) [inline]
```

Definition at line 32 of file algo.h.

Referenced by any().

### 5.2.3.2 \_\_any() [2/2]

```
template<class RandomAccessIterator , class T >
bool Rcpp::__any (
    RandomAccessIterator __first,
    RandomAccessIterator __last,
    const T & __val,
    std::random_access_iterator_tag ) [inline]
```

Definition at line 39 of file algo.h.

**5.2.3.3 `__any_if()`** [1/2]

```
template<class InputIterator , class Predicate >
bool Rcpp::__any_if (
    InputIterator first,
    InputIterator last,
    Predicate pred,
    std::input_iterator_tag ) [inline]
```

Definition at line 98 of file algo.h.

Referenced by `any_if()`.

**5.2.3.4 `__any_if()`** [2/2]

```
template<class RandomAccessIterator , class Predicate >
bool Rcpp::__any_if (
    RandomAccessIterator __first,
    RandomAccessIterator __last,
    Predicate __pred,
    std::random_access_iterator_tag ) [inline]
```

Definition at line 105 of file algo.h.

**5.2.3.5 `all()`**

```
template<bool NA, typename T >
sugar::All<NA,T> Rcpp::all (
    const Rcpp::VectorBase< LGLSXP, NA, T > & t ) [inline]
```

Definition at line 84 of file all.h.

Referenced by `ls()`.

**5.2.3.6 `any()`** [1/2]

```
template<bool NA, typename T >
sugar::Any<NA,T> Rcpp::any (
    const Rcpp::VectorBase< LGLSXP, NA, T > & t ) [inline]
```

Definition at line 81 of file any.h.

### 5.2.3.7 any() [2/2]

```
template<class InputIterator , class T >
bool Rcpp::any (
    InputIterator first,
    InputIterator last,
    const T & value ) [inline]
```

stl like algorithm to identify if any of the objects in the range is equal to the value

Definition at line 89 of file algo.h.

References `__any()`.

Referenced by `Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::operator result_type()`, and `Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::operator result_type()`.

Here is the call graph for this function:



### 5.2.3.8 any\_if()

```
template<class InputIterator , class Predicate >
bool Rcpp::any_if (
    InputIterator first,
    InputIterator last,
    Predicate pred ) [inline]
```

stl-like algorithm to identify if the predicate is true for any of the objects in the range

Definition at line 155 of file algo.h.

References `__any_if()`.

Here is the call graph for this function:



### 5.2.3.9 as() [1/2]

```
template<typename T >
T Rcpp::as (
    SEXP x )
```

Generic converted from SEXP to the typename. T can be any type that has a constructor taking a SEXP, which is the case for all our RObject and derived classes.

If it is not possible to add the SEXP constructor, e.g you don't control the type, you can specialize the as template to perform the requested conversion

This is used for example in Environment, so that for example the code below will work as long as there is a way to as<> the Foo type

```
Environment x = ... ; // some environment Foo y = x["bla"] ; // if as<Foo> makes sense then this works !!
```

Definition at line 151 of file as.h.

Referenced by CppProperty\_GetMethod\_SetMethod< Class, PROP >::set(), CppProperty\_GetConstMethod\_SetMethod< Class, PROP >::set(), CppProperty\_GetMethod\_SetPointer< Class, PROP >::set(), CppProperty\_GetConstMethod\_SetPointer< Class, PROP >::set(), CppProperty\_GetPointer\_SetMethod< Class, PROP >::set(), and CppProperty\_GetPointer\_SetPointer< Class, PROP >::set().

### 5.2.3.10 as() [2/2]

```
template<>
SEXP Rcpp::as (
    SEXP x ) [inline]
```

Definition at line 164 of file as.h.

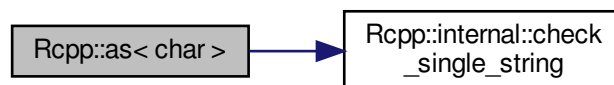
### 5.2.3.11 as< char >()

```
template<>
char Rcpp::as< char > (
    SEXP x ) [inline]
```

Definition at line 155 of file as.h.

References Rcpp::internal::check\_single\_string().

Here is the call graph for this function:



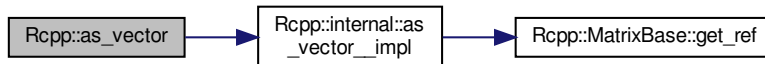
### 5.2.3.12 as\_vector()

```
template<int RTYPE, bool NA, typename T >
Rcpp::Vector<RTYPE> Rcpp::as_vector (
    const MatrixBase< RTYPE, NA, T > & t ) [inline]
```

Definition at line 55 of file as\_vector.h.

References Rcpp::internal::as\_vector\_\_impl().

Here is the call graph for this function:



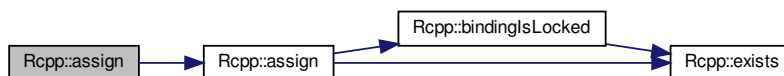
### 5.2.3.13 assign() [1/3]

```
bool Rcpp::assign (
    const std::string & name,
    const Shield< SEXP > & x ) const
```

Definition at line 214 of file Environment.h.

References assign().

Here is the call graph for this function:



### 5.2.3.14 assign() [2/3]

```
template<typename WRAPPABLE >
bool Rcpp::assign (
    const std::string & name,
    const WRAPPABLE & x ) const
```

wrap and assign. If there is a wrap method taking an object of WRAPPABLE type, then it is wrapped and the corresponding SEXP is assigned in the environment

## Parameters

<i>name</i>	name of the object to assign
<i>x</i>	wrappable object. anything that has a wrap( WRAPPABLE ) is fine

**5.2.3.15 assign()** [3/3]

```
bool Rcpp::assign (
    const std::string & name,
    SEXP x ) const
```

Attempts to assign *x* to *name* in this environment

## Parameters

<i>name</i>	name of the object to assign
<i>x</i>	object to assign

## Returns

true if the assign was successfull see ?bindingsLocked

## Exceptions

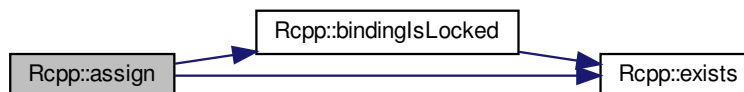
<i>binding_is_locked</i>	if the binding is locked
--------------------------	--------------------------

Definition at line 207 of file Environment.h.

References bindingsLocked(), and exists().

Referenced by assign().

Here is the call graph for this function:



### 5.2.3.16 bare\_as()

```
template<typename T >
traits::remove_const_and_reference<T>::type Rcpp::bare_as (
    SEXP x ) [inline]
```

Definition at line 160 of file as.h.

### 5.2.3.17 base\_env()

```
static Environment_Impl Rcpp::base_env ( ) [static]
```

#### Returns

the base environment. See ?baseenv

Definition at line 346 of file Environment.h.

Referenced by Rcpp::attributes::checkRSignature(), Rcpp::attributes::createDirectory(), message(), Rcpp::attributes::removeFile(), and Rcpp::attributes::showWarning().

### 5.2.3.18 base\_namespace()

```
static Environment_Impl Rcpp::base_namespace ( ) [static]
```

#### Returns

the base namespace. See ?baseenv

Definition at line 353 of file Environment.h.

### 5.2.3.19 bindingIsActive()

```
bool Rcpp::bindingIsActive (
    const std::string & name ) const
```

#### Parameters

<i>name</i>	name of a binding
-------------	-------------------

**Returns**

true if the binding is active in this environment see ?bindingsActive

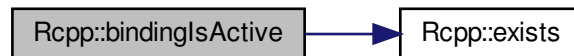
**Exceptions**

<i>no_such_binding</i>	if there is no such binding in this environment
------------------------	---

Definition at line 315 of file Environment.h.

References exists().

Here is the call graph for this function:

**5.2.3.20 bindingsLocked()**

```

bool Rcpp::bindingIsLocked (
    const std::string & name ) const
  
```

**Parameters**

<i>name</i>	name of a potential binding
-------------	-----------------------------

**Returns**

true if the binding is locked in this environment see ?bindingsLocked

**Exceptions**

<i>no_such_binding</i>	if there is no such binding in this environment
------------------------	---

Definition at line 300 of file Environment.h.

References exists().



Referenced by `assign()`, and `remove()`.

Here is the call graph for this function:



### 5.2.3.21 `checkUserInterrupt()`

```
void Rcpp::checkUserInterrupt ( ) [inline]
```

Definition at line 59 of file `Interrupt.h`.

### 5.2.3.22 `clamp()`

```
template<int RTYPE, bool NA, typename T >  
sugar::Clamp_Primitive_Vector_Primitive<RTYPE,NA,T> Rcpp::clamp (  
    typename Rcpp::traits::storage_type< RTYPE >::type lhs,  
    const Rcpp::VectorBase< RTYPE, NA, T > & vec,  
    typename Rcpp::traits::storage_type< RTYPE >::type rhs ) [inline]
```

Definition at line 84 of file `clamp.h`.

References `Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref()`.

Here is the call graph for this function:



### 5.2.3.23 clone()

```
template<typename T >
T Rcpp::clone (
    const T & object )
```

Definition at line 33 of file clone.h.

Referenced by Rcpp::Nullable< T >::clone(), and sample().

### 5.2.3.24 col()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
sugar::Col<RTYPE, LHS_NA, LHS_T> Rcpp::col (
    const Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T > & lhs ) [inline]
```

Definition at line 55 of file col.h.

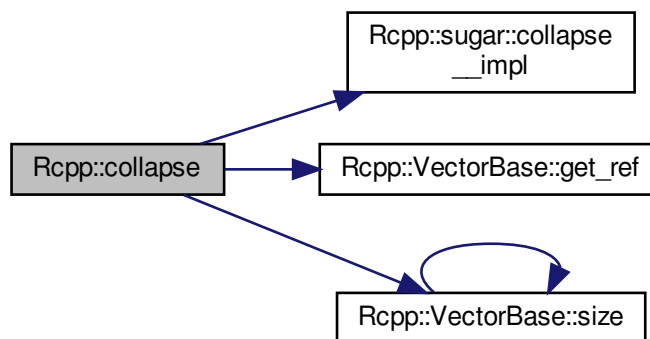
### 5.2.3.25 collapse()

```
template<bool NA, typename T >
String Rcpp::collapse (
    const VectorBase< STRSXP, NA, T > & vec ) [inline]
```

Definition at line 41 of file collapse.h.

References Rcpp::sugar::collapse\_\_impl(), Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref(), and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



### 5.2.3.26 colMeans()

```
template<int RTYPE, bool NA, typename T >
sugar::detail::ColMeansReturn<RTYPE>::type Rcpp::colMeans (
    const MatrixBase< RTYPE, NA, T > & x,
    bool na_rm = false ) [inline]
```

Definition at line 951 of file rowSums.h.

### 5.2.3.27 colnames()

```
internal::DimNameProxy Rcpp::colnames (
    SEXP x ) [inline]
```

Definition at line 213 of file Matrix.h.

### 5.2.3.28 colSums()

```
template<int RTYPE, bool NA, typename T >
sugar::detail::ColSumsReturn<RTYPE>::type Rcpp::colSums (
    const MatrixBase< RTYPE, NA, T > & x,
    bool na_rm = false ) [inline]
```

Definition at line 933 of file rowSums.h.

### 5.2.3.29 converter()

```
template<typename FROM , typename TO >
void Rcpp::converter (
    const char * from,
    const char * to,
    TO(*) (FROM) fun,
    const char * docstring = 0 )
```

Definition at line 377 of file Module.h.

### 5.2.3.30 cummax() [1/2]

```
template<bool NA, typename T >  
sugar::Cummax<INTSXP, NA, T> Rcpp::cummax (  
    const VectorBase< INTSXP, NA, T > & t ) [inline]
```

Definition at line 59 of file cummax.h.

### 5.2.3.31 cummax() [2/2]

```
template<bool NA, typename T >  
sugar::Cummax<REALSXP, NA, T> Rcpp::cummax (  
    const VectorBase< REALSXP, NA, T > & t ) [inline]
```

Definition at line 64 of file cummax.h.

### 5.2.3.32 cummin() [1/2]

```
template<bool NA, typename T >  
sugar::Cummin<INTSXP, NA, T> Rcpp::cummin (  
    const VectorBase< INTSXP, NA, T > & t ) [inline]
```

Definition at line 59 of file cummin.h.

### 5.2.3.33 cummin() [2/2]

```
template<bool NA, typename T >  
sugar::Cummin<REALSXP, NA, T> Rcpp::cummin (  
    const VectorBase< REALSXP, NA, T > & t ) [inline]
```

Definition at line 64 of file cummin.h.

### 5.2.3.34 cumprod() [1/3]

```
template<bool NA, typename T >  
sugar::Cumprod<CPLXSXP, NA, T> Rcpp::cumprod (  
    const VectorBase< CPLXSXP, NA, T > & t ) [inline]
```

Definition at line 69 of file cumprod.h.

**5.2.3.35 cumprod()** [2/3]

```
template<bool NA, typename T >
sugar::Cumprod<INTSXP, NA, T> Rcpp::cumprod (
    const VectorBase< INTSXP, NA, T > & t ) [inline]
```

Definition at line 59 of file cumprod.h.

**5.2.3.36 cumprod()** [3/3]

```
template<bool NA, typename T >
sugar::Cumprod<REALSXP, NA, T> Rcpp::cumprod (
    const VectorBase< REALSXP, NA, T > & t ) [inline]
```

Definition at line 64 of file cumprod.h.

**5.2.3.37 cumsum()** [1/2]

```
template<bool NA, typename T >
sugar::Cumsum<INTSXP, NA, T> Rcpp::cumsum (
    const VectorBase< INTSXP, NA, T > & t ) [inline]
```

Definition at line 59 of file cumsum.h.

**5.2.3.38 cumsum()** [2/2]

```
template<bool NA, typename T >
sugar::Cumsum<REALSXP, NA, T> Rcpp::cumsum (
    const VectorBase< REALSXP, NA, T > & t ) [inline]
```

Definition at line 64 of file cumsum.h.

**5.2.3.39 detzcode()**

```
static int_fast32_t Rcpp::detzcode (
    const char *const codep ) [static]
```

Definition at line 482 of file date.cpp.

Referenced by tzload().

#### 5.2.3.40 detzcode64()

```
static int_fast64_t Rcpp::detzcode64 (
    const char *const codep ) [static]
```

Definition at line 489 of file `date.cpp`.

Referenced by `tzload()`.

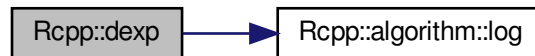
#### 5.2.3.41 dexp()

```
template<bool NA, typename T >
stats::D1<REALSXP,NA,T> Rcpp::dexp (
    const Rcpp::VectorBase< REALSXP, NA, T > & x,
    double shape,
    bool log = false ) [inline]
```

Definition at line 81 of file `exp.h`.

References `Rcpp::algorithm::log()`.

Here is the call graph for this function:



#### 5.2.3.42 diag()

```
template<typename T >
sugar::diag_result_type_trait<T>::type Rcpp::diag (
    const T & t ) [inline]
```

Definition at line 83 of file `diag.h`.

Referenced by `Rcpp::MatrixBase< RTYPE, na, MATRIX >::eye()`, `lower_tri()`, and `upper_tri()`.

**5.2.3.43 diff()** [1/2]

```
template<bool LHS_NA, typename LHS_T >
sugar::Diff<INTSXP,LHS_NA,LHS_T> Rcpp::diff (
    const VectorBase< INTSXP, LHS_NA, LHS_T > & lhs ) [inline]
```

Definition at line 124 of file diff.h.

Referenced by compileAttributes(), and Rcpp::sugar::Diff< RTYPE, false, LHS\_T >::operator[]().

**5.2.3.44 diff()** [2/2]

```
template<bool LHS_NA, typename LHS_T >
sugar::Diff<REALSXP,LHS_NA,LHS_T> Rcpp::diff (
    const VectorBase< REALSXP, LHS_NA, LHS_T > & lhs ) [inline]
```

Definition at line 131 of file diff.h.

**5.2.3.45 differ\_by\_repeat()**

```
static int Rcpp::differ_by_repeat (
    const time_t t1,
    const time_t t0 ) [static]
```

Definition at line 496 of file date.cpp.

References SECSPERREPEAT, SECSPERREPEAT\_BITS, TYPE\_BIT, TYPE\_INTEGRAL, and TYPE\_SIGNED.

Referenced by tzload().

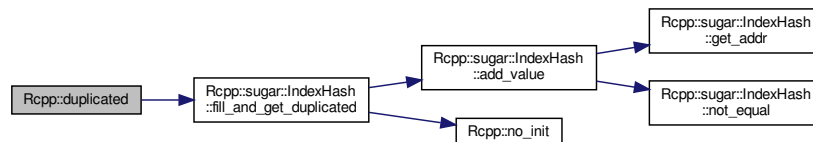
**5.2.3.46 duplicated()**

```
template<int RTYPE, bool NA, typename T >
LogicalVector Rcpp::duplicated (
    const VectorBase< RTYPE, NA, T > & x ) [inline]
```

Definition at line 28 of file duplicated.h.

References Rcpp::sugar::IndexHash< RTYPE >::fill\_and\_get\_duplicated().

Here is the call graph for this function:



### 5.2.3.47 `empty_env()`

```
static Environment_Impl Rcpp::empty_env ( ) [static]
```

#### Returns

The empty environment. See `?emptyenv`

Definition at line 339 of file `Environment.h`.

### 5.2.3.48 `Environment_Impl()` [1/3]

```
Rcpp::Environment_Impl (
    const std::string & name )
```

Gets the environment associated with the given name

#### Parameters

<i>name</i>	name of the environment, e.g "package:Rcpp"
-------------	---

### 5.2.3.49 `Environment_Impl()` [2/3]

```
Rcpp::Environment_Impl (
    int pos )
```

Gets the environment in the given position of the search path

#### Parameters

<i>pos</i>	(1-based) position of the environment, e.g pos=1 gives the global environment
------------	---

### 5.2.3.50 `Environment_Impl()` [3/3]

```
Rcpp::Environment_Impl (
    SEXP x )
```

wraps the given environment



if the SEXP is not an environment, and exception is thrown

Definition at line 57 of file Environment.h.

Referenced by namespace\_env(), new\_child(), and parent().

### 5.2.3.51 eval() [1/2]

```
SEXP Rcpp::eval ( ) const
```

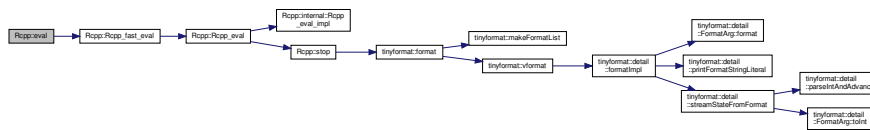
eval this call in the global environment

Definition at line 135 of file Language.h.

References Rcpp\_fast\_eval().

Referenced by Rcpp::attributes::checkRSignature().

Here is the call graph for this function:



### 5.2.3.52 eval() [2/2]

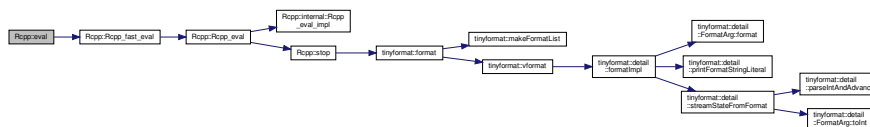
```
SEXP Rcpp::eval (
    SEXP env ) const
```

eval this call in the requested environment

Definition at line 142 of file Language.h.

References Rcpp\_fast\_eval().

Here is the call graph for this function:



### 5.2.3.53 exists()

```
bool Rcpp::exists (
    const std::string & name ) const
```

Indicates if an object called name exists in the environment

**Parameters**

<i>name</i>	name of the object
-------------	--------------------

**Returns**

true if the object exists in the environment

Definition at line 190 of file Environment.h.

Referenced by `assign()`, `bindingsActive()`, `bindingsLocked()`, `Rcpp::attributes::ExportsGenerator::commit()`, `Rcpp::attributes::createDirectory()`, `Rcpp::attributes::ExportsGenerator::ExportsGenerator()`, `lockBinding()`, `remove()`, `Rcpp::attributes::removeFile()`, and `unlockBinding()`.

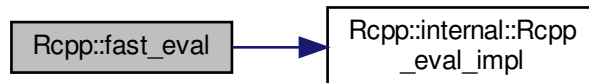
**5.2.3.54 fast\_eval() [1/2]**

```
SEXP Rcpp::fast_eval ( ) const
```

Definition at line 146 of file Language.h.

References `Rcpp::internal::Rcpp_eval_impl()`.

Here is the call graph for this function:



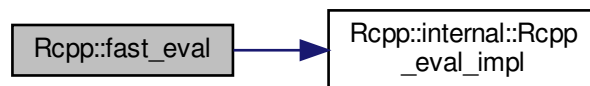
### 5.2.3.55 `fast_eval()` [2/2]

```
SEXP Rcpp::fast_eval (
    SEXP env ) const
```

Definition at line 149 of file Language.h.

References `Rcpp::internal::Rcpp_eval_impl()`.

Here is the call graph for this function:



### 5.2.3.56 `finalizer_wrapper()`

```
template<typename T , void Finalizer>
void Rcpp::finalizer_wrapper (
    SEXP p )
```

Definition at line 35 of file XPtr.h.

References `DEMANGLE`, and `RCPP_DEBUG_3`.

### 5.2.3.57 `find()` [1/2]

```
SEXP Rcpp::find (
    const std::string & name ) const
```

Get an object from the environment or one of its parents

#### Parameters

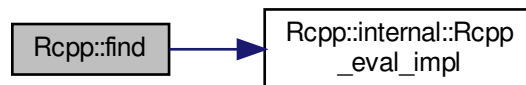
<i>name</i>	name of the object
-------------	--------------------

Definition at line 145 of file Environment.h.

References Rcpp::internal::Rcpp\_eval\_impl().

Referenced by Rcpp::attributes::checkRSignature(), Rcpp::attributes::CppExportsIncludeGenerator::commit(), and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::get\_indices().

Here is the call graph for this function:



### 5.2.3.58 find() [2/2]

```
SEXP Rcpp::find (
    Symbol name ) const
```

Get an object from the environment or one of its parents

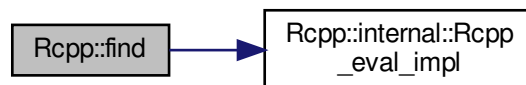
#### Parameters

<i>name</i>	symbol name to call
-------------	---------------------

Definition at line 165 of file Environment.h.

References Rcpp::internal::Rcpp\_eval\_impl().

Here is the call graph for this function:





**Parameters**

<i>name</i>	name of the object
-------------	--------------------

**Returns**

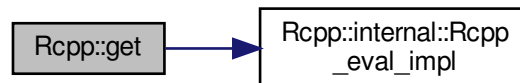
a SEXP (possibly R\_NilValue)

Definition at line 103 of file Environment.h.

References Rcpp::internal::Rcpp\_eval\_impl().

Referenced by Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator SEXP(), Rcpp::AttributeProxyPolicy< CLASS >::const\_AttributeProxy::operator SEXP(), Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::operator SEXP(), Rcpp::TagProxyPolicy< XPtrClass >::const\_TagProxy::operator SEXP(), Rcpp::AttributeProxyPolicy< CLASS >::const\_AttributeProxy::operator T(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator T(), Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding::operator T(), Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator T(), Rcpp::DottedPairProxyPolicy< CLASS >::const\_DottedPairProxy::operator T(), Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator T(), Rcpp::FieldProxyPolicy< CLASS >::const\_FieldProxy::operator T(), Rcpp::NamesProxyPolicy< CLASS >::const\_NamesProxy::operator T(), Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator T(), and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::operator+=().

Here is the call graph for this function:

**5.2.3.62 get() [2/2]**

```
SEXP Rcpp::get (
    Symbol name ) const
```

Get an object from the environment

**Parameters**

<i>name</i>	symbol name to call
-------------	---------------------

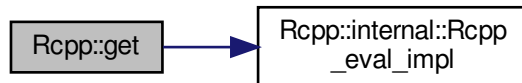
**Returns**

a SEXP (possibly R\_NilValue)

Definition at line 124 of file Environment.h.

References Rcpp::internal::Rcpp\_eval\_impl().

Here is the call graph for this function:

**5.2.3.63 get\_return\_type()**

```
template<typename RESULT_TYPE >  
std::string Rcpp::get_return_type ( ) [inline]
```

Definition at line 42 of file get\_return\_type.h.

**5.2.3.64 get\_return\_type< Rcpp::CharacterVector >()**

```
template<>  
std::string Rcpp::get_return_type< Rcpp::CharacterVector > ( ) [inline]
```

Definition at line 74 of file get\_return\_type.h.

**5.2.3.65 get\_return\_type< Rcpp::ExpressionVector >()**

```
template<>  
std::string Rcpp::get_return_type< Rcpp::ExpressionVector > ( ) [inline]
```

Definition at line 66 of file get\_return\_type.h.

### 5.2.3.66 `get_return_type< Rcpp::IntegerVector >()`

```
template<>  
std::string Rcpp::get_return_type< Rcpp::IntegerVector > ( ) [inline]
```

Definition at line 54 of file `get_return_type.h`.

### 5.2.3.67 `get_return_type< Rcpp::List >()`

```
template<>  
std::string Rcpp::get_return_type< Rcpp::List > ( ) [inline]
```

Definition at line 70 of file `get_return_type.h`.

### 5.2.3.68 `get_return_type< Rcpp::NumericVector >()`

```
template<>  
std::string Rcpp::get_return_type< Rcpp::NumericVector > ( ) [inline]
```

Definition at line 58 of file `get_return_type.h`.

### 5.2.3.69 `get_return_type< Rcpp::RawVector >()`

```
template<>  
std::string Rcpp::get_return_type< Rcpp::RawVector > ( ) [inline]
```

Definition at line 62 of file `get_return_type.h`.

### 5.2.3.70 `get_return_type< SEXP >()`

```
template<>  
std::string Rcpp::get_return_type< SEXP > ( ) [inline]
```

Definition at line 50 of file `get_return_type.h`.



### 5.2.3.71 `get_return_type< void_type >()`

```
template<>
std::string Rcpp::get_return_type< void_type > ( ) [inline]
```

Definition at line 46 of file `get_return_type.h`.

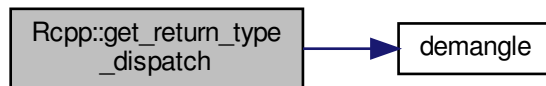
### 5.2.3.72 `get_return_type_dispatch()` [1/2]

```
template<typename RESULT_TYPE >
std::string Rcpp::get_return_type_dispatch (
    Rcpp::traits::false_type ) [inline]
```

Definition at line 30 of file `get_return_type.h`.

References `demangle()`.

Here is the call graph for this function:



### 5.2.3.73 `get_return_type_dispatch()` [2/2]

```
template<typename RESULT_TYPE >
std::string Rcpp::get_return_type_dispatch (
    Rcpp::traits::true_type ) [inline]
```

Definition at line 34 of file `get_return_type.h`.

References `demangle()`.

Here is the call graph for this function:



### 5.2.3.74 `getnum()`

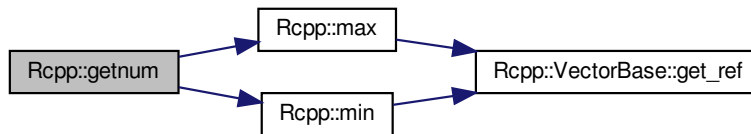
```
static const char * Rcpp::getnum (
    const char * strp,
    int *const nump,
    const int min,
    const int max ) [static]
```

Definition at line 568 of file `date.cpp`.

References `is_digit`, `max()`, and `min()`.

Referenced by `getrule()`, and `getsecs()`.

Here is the call graph for this function:



### 5.2.3.75 `getoffset()`

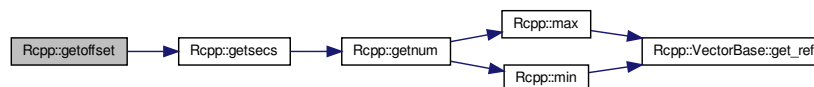
```
static const char* Rcpp::getoffset (
    const char * strp,
    int_fast32_t *const offsetp ) [static]
```

Definition at line 521 of file `date.cpp`.

References `getsecs()`.

Referenced by `tzparse()`.

Here is the call graph for this function:



### 5.2.3.76 getqzname()

```
static const char* Rcpp::getqzname (
    const char * strp,
    const int delim ) [static]
```

Definition at line 513 of file date.cpp.

Referenced by tzparse().

### 5.2.3.77 getrule()

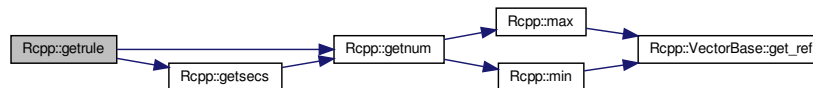
```
static const char * Rcpp::getrule (
    const char * strp,
    struct rule *const rulep ) [static]
```

Definition at line 587 of file date.cpp.

References DAY\_OF\_YEAR, DAYSPERYEAR, DAYSPERNYEAR, DAYSPERWEEK, getnum(), getsecs(), is\_digit, JULIAN\_DAY, MONSPERYEAR, MONTH\_NTH\_DAY\_OF\_WEEK, Rcpp::rule::r\_day, Rcpp::rule::r\_mon, Rcpp::rule::r←\_time, Rcpp::rule::r\_type, Rcpp::rule::r\_week, and SECSPERHOUR.

Referenced by tzparse().

Here is the call graph for this function:



### 5.2.3.78 getsecs()

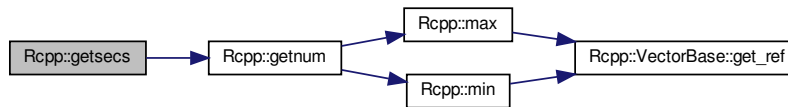
```
static const char * Rcpp::getsecs (
    const char * strp,
    int_fast32_t * secp ) [static]
```

Definition at line 537 of file date.cpp.

References DAYSPERWEEK, getnum(), HOURSPERDAY, MINSPERHOUR, SECSPERHOUR, and SECSPERMIN.

Referenced by getoffset(), and getrule().

Here is the call graph for this function:



### 5.2.3.79 getzname()

```
static const char* Rcpp::getzname (
    const char * strp ) [static]
```

Definition at line 504 of file date.cpp.

References `is_digit`.

Referenced by `tzparse()`.

### 5.2.3.80 global\_env()

```
static Environment_Impl Rcpp::global_env ( ) [static]
```

#### Returns

the global environment. See `?globalenv`

Definition at line 332 of file Environment.h.

### 5.2.3.81 gmtime\_()

```
struct tm * Rcpp::gmtime_ (
    const time_t *const x ) [inline]
```

Definition at line 136 of file routines.h.

References `GET_CALLABLE`, and `tm`.

Referenced by `registerFunctions()`, `Rcpp::Date::update_tm()`, and `Rcpp::Datetime::update_tm()`.

### 5.2.3.82 gmtime()

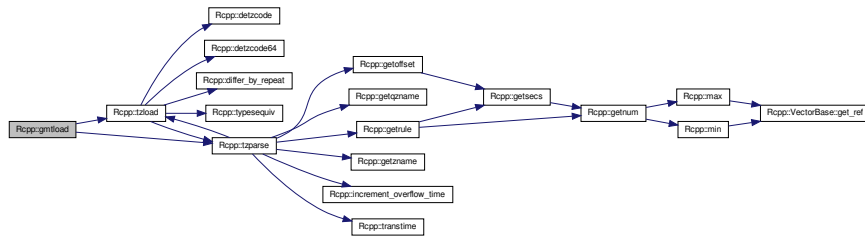
```
static void Rcpp::gmtime (
    struct state *const sp ) [static]
```

Definition at line 1356 of file date.cpp.

References `gmt`, `tzload()`, and `tzparse()`.

Referenced by `gmtsub()`.

Here is the call graph for this function:



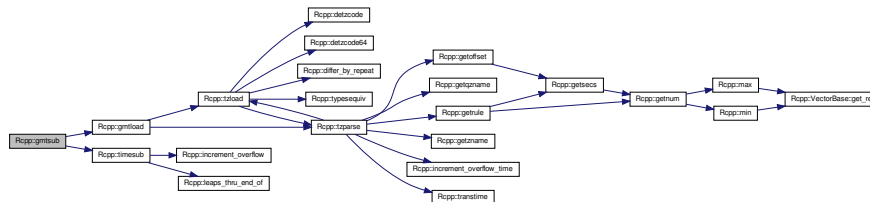
### 5.2.3.83 gmtime\_sub()

```
static struct tm* Rcpp::gmtime_sub (
    const time_t *const timep,
    const int_fast32_t offset,
    struct tm *const tmp ) [static]
```

Definition at line 1365 of file date.cpp.

References `gmt_is_set`, `gmtime()`, `gmtimeptr`, `timesub()`, and `tm`.

Here is the call graph for this function:



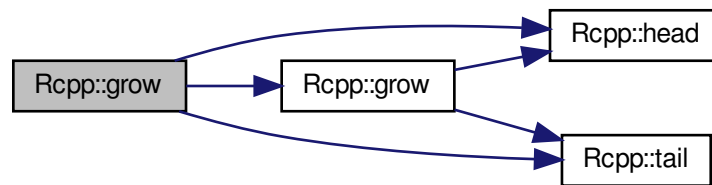
**5.2.3.84** `grow()` [1/3]

```
SEXP Rcpp::grow (
    const char * head,
    SEXP tail ) [inline]
```

Definition at line 68 of file `grow.h`.

References `grow()`, `head()`, and `tail()`.

Here is the call graph for this function:

**5.2.3.85** `grow()` [2/3]

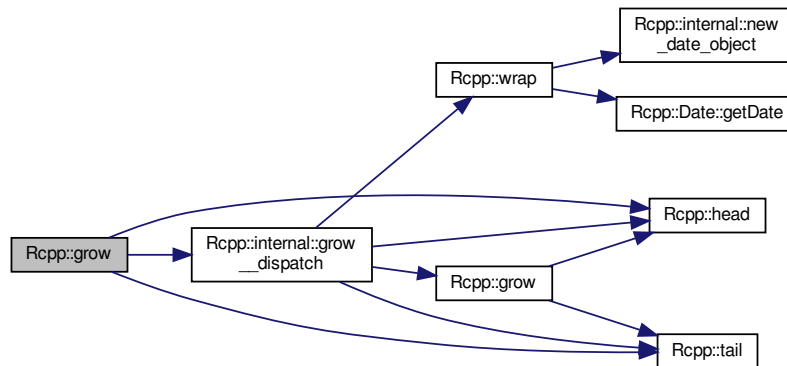
```
template<typename T >
SEXP Rcpp::grow (
    const T & head,
    SEXP tail )
```

grows a pairlist. First wrap the head into a SEXP, then grow the tail pairlist

Definition at line 63 of file `grow.h`.

References `Rcpp::internal::grow__dispatch()`, `head()`, and `tail()`.

Here is the call graph for this function:



### 5.2.3.86 `grow()` [3/3]

```

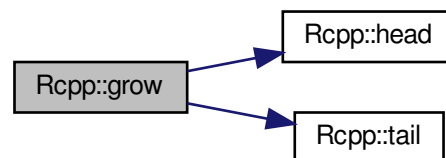
SEXP Rcpp::grow (
    SEXP head,
    SEXP tail ) [inline]
  
```

Definition at line 34 of file `grow.h`.

References `head()`, and `tail()`.

Referenced by `grow()`, `Rcpp::internal::grow__dispatch()`, `Rcpp::DottedPairImpl< CLASS >::insert()`, `Rcpp::DottedPairImpl< CLASS >::push_back()`, and `Rcpp::DottedPairImpl< CLASS >::push_front()`.

Here is the call graph for this function:



### 5.2.3.87 head()

```
template<int RTYPE, bool NA, typename T >
sugar::Head<RTYPE,NA,T> Rcpp::head (
    const VectorBase< RTYPE, NA, T > & t,
    R_xlen_t n ) [inline]
```

Definition at line 53 of file head.h.

Referenced by grow(), and Rcpp::internal::grow\_\_dispatch().

### 5.2.3.88 ifelse() [1/7]

```
template<bool COND_NA, typename COND_T >
sugar::IfElse_Primitive_Primitive< LGLSXP,COND_NA,COND_T > Rcpp::ifelse (
    const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > & cond,
    bool lhs,
    bool rhs ) [inline]
```

Definition at line 419 of file ifelse.h.

### 5.2.3.89 ifelse() [2/7]

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
sugar::IfElse< RTYPE,COND_NA,COND_T,LHS_NA,LHS_T,RHS_NA,RHS_T > Rcpp::ifelse (
    const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > & cond,
    const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 342 of file ifelse.h.

### 5.2.3.90 ifelse() [3/7]

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
sugar::IfElse_Vector_Primitive< RTYPE,COND_NA,COND_T,RHS_NA,RHS_T > Rcpp::ifelse (
    const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > & cond,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & lhs,
    typename traits::storage_type< RTYPE >::type rhs ) [inline]
```

Definition at line 371 of file ifelse.h.



**5.2.3.91 ifelse()** [4/7]

```
template<bool COND_NA, typename COND_T >
sugar::IfElse_Primitive_Primitive< REALSXP,COND_NA,COND_T > Rcpp::ifelse (
    const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > & cond,
    double lhs,
    double rhs ) [inline]
```

Definition at line 383 of file ifelse.h.

**5.2.3.92 ifelse()** [5/7]

```
template<bool COND_NA, typename COND_T >
sugar::IfElse_Primitive_Primitive< INTSXP,COND_NA,COND_T > Rcpp::ifelse (
    const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > & cond,
    int lhs,
    int rhs ) [inline]
```

Definition at line 395 of file ifelse.h.

**5.2.3.93 ifelse()** [6/7]

```
template<bool COND_NA, typename COND_T >
sugar::IfElse_Primitive_Primitive< CPLXSXP,COND_NA,COND_T > Rcpp::ifelse (
    const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > & cond,
    Rcomplex lhs,
    Rcomplex rhs ) [inline]
```

Definition at line 407 of file ifelse.h.

**5.2.3.94 ifelse()** [7/7]

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
sugar::IfElse_Primitive_Vector< RTYPE,COND_NA,COND_T,RHS_NA,RHS_T > Rcpp::ifelse (
    const Rcpp::VectorBase< LGLSXP, COND_NA, COND_T > & cond,
    typename traits::storage_type< RTYPE >::type lhs,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 357 of file ifelse.h.

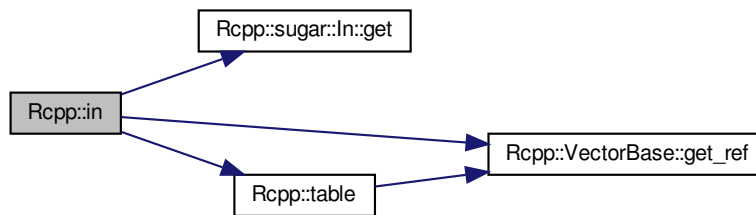
### 5.2.3.95 in()

```
template<int RTYPE, bool NA, typename T , bool RHS_NA, typename RHS_T >
LogicalVector Rcpp::in (
    const VectorBase< RTYPE, NA, T > & x,
    const VectorBase< RTYPE, RHS_NA, RHS_T > & table ) [inline]
```

Definition at line 77 of file unique.h.

References `Rcpp::sugar::In< RTYPE, TABLE_T >::get()`, `Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref()`, and `table()`.

Here is the call graph for this function:



### 5.2.3.96 increment\_overflow()

```
static int Rcpp::increment_overflow (
    int *const ip,
    int j ) [static]
```

Definition at line 453 of file date.cpp.

Referenced by `timesub()`.

### 5.2.3.97 increment\_overflow\_time()

```
static int Rcpp::increment_overflow_time (
    time_t * tp,
    int_fast32_t j ) [static]
```

Definition at line 468 of file date.cpp.

References `time_t_max`, `time_t_min`, and `TYPE_SIGNED`.

Referenced by `tzparse()`.

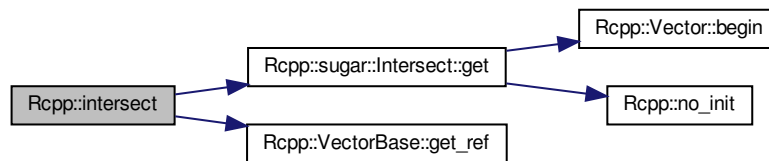
**5.2.3.98 intersect()**

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Vector<RTYPE> Rcpp::intersect (
    const VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 171 of file setdiff.h.

References `Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get()`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref()`.

Here is the call graph for this function:

**5.2.3.99 is()**

```
template<typename T >
bool Rcpp::is (
    SEXP x )
```

identify if an x can be seen as the T type

example: `bool is_list = is<List>( x ) ;`

Definition at line 53 of file is.h.

Referenced by `R_CPP_API_CLASS()`.

**5.2.3.100 is\_false()**

```
template<bool NA, typename T >
bool Rcpp::is_false (
    const Rcpp::sugar::SingleLogicalResult< NA, T > & x ) [inline]
```

Definition at line 33 of file is.h.

### 5.2.3.101 is\_finite()

```
template<int RTYPE, bool NA, typename T >  
sugar::IsFinite<RTYPE,NA,T> Rcpp::is_finite (  
    const Rcpp::VectorBase< RTYPE, NA, T > & t ) [inline]
```

Definition at line 49 of file is\_finite.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Here is the call graph for this function:



### 5.2.3.102 is\_infinite()

```
template<int RTYPE, bool NA, typename T >  
sugar::IsInfinite<RTYPE,NA,T> Rcpp::is_infinite (  
    const Rcpp::VectorBase< RTYPE, NA, T > & t ) [inline]
```

Definition at line 49 of file is\_infinite.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Here is the call graph for this function:



**5.2.3.103 is\_na()** [1/6]

```
sugar::IsNa_Vector_is_na<oldDatetimeVector> Rcpp::is_na (  
    const oldDatetimeVector & x ) [inline]
```

Definition at line 95 of file is\_na.h.

**5.2.3.104 is\_na()** [2/6]

```
sugar::IsNa_Vector_is_na<oldDateVector> Rcpp::is_na (  
    const oldDateVector & x ) [inline]
```

Definition at line 99 of file is\_na.h.

**5.2.3.105 is\_na()** [3/6]

```
template<bool NA, typename T >  
bool Rcpp::is_na (  
    const Rcpp::sugar::SingleLogicalResult< NA, T > & x ) [inline]
```

Definition at line 38 of file is.h.

References [is\\_na\(\)](#).

Here is the call graph for this function:



**5.2.3.106 is\_na()** [4/6]

```
template<int RTYPE, bool NA, typename T >
sugar::IsNa<RTYPE,NA,T> Rcpp::is_na (
    const Rcpp::VectorBase< RTYPE, NA, T > & t ) [inline]
```

Definition at line 91 of file is\_na.h.

Referenced by is\_na(), Rcpp::sugar::na\_omit\_impl(), Rcpp::sugar::Median< RTYPE, NA, T, NA\_RM >::operator result\_type(), Rcpp::sugar::Median< STRSXP, NA, T, NA\_RM >::operator result\_type(), Rcpp::algorithm::helpers::log↔::operator(), Rcpp::algorithm::helpers::exp::operator(), and Rcpp::algorithm::helpers::sqrt::operator().

**5.2.3.107 is\_na()** [5/6]

```
sugar::IsNa_Vector_is_na<NumericVector> Rcpp::is_na (
    newDatetimeVector & x ) [inline]
```

Definition at line 103 of file is\_na.h.

**5.2.3.108 is\_na()** [6/6]

```
sugar::IsNa_Vector_is_na<NumericVector> Rcpp::is_na (
    newDateVector & x ) [inline]
```

Definition at line 106 of file is\_na.h.

**5.2.3.109 is\_nan()**

```
template<int RTYPE, bool NA, typename T >
sugar::IsNaN<RTYPE,NA,T> Rcpp::is_nan (
    const Rcpp::VectorBase< RTYPE, NA, T > & t ) [inline]
```

Definition at line 49 of file is\_nan.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Here is the call graph for this function:



### 5.2.3.110 is\_true()

```
template<bool NA, typename T >
bool Rcpp::is_true (
    const Rcpp::sugar::SingleLogicalResult< NA, T > & x ) [inline]
```

Definition at line 28 of file is.h.

### 5.2.3.111 is\_user\_database()

```
bool Rcpp::is_user_database ( ) const
```

Indicates if this is a user defined database.

Definition at line 324 of file Environment.h.

Referenced by ls().

### 5.2.3.112 isLocked()

```
bool Rcpp::isLocked ( ) const
```

#### Returns

true if this environment is locked see ?environmentIsLocked for details of what this means

Definition at line 233 of file Environment.h.

### 5.2.3.113 Language\_Impl() [1/5]

```
Rcpp::Language_Impl ( )
```

Definition at line 42 of file Language.h.

### 5.2.3.114 Language\_Impl() [2/5]

```
Rcpp::Language_Impl (
    const Function & function ) [explicit]
```

Creates a call to the function

**Parameters**

<i>function</i>	function to call
-----------------	------------------

Definition at line 84 of file Language.h.

**5.2.3.115 Language\_Impl() [3/5]**

```
Rcpp::Language_Impl (
    const std::string & symbol ) [explicit]
```

Creates a call using the given symbol as the function name

**Parameters**

<i>symbol</i>	symbol name to call
---------------	---------------------

Language( "rnorm" ) makes a SEXP similar to this (expressed in [R](#))

```
as.call( as.list( as.name( "rnorm" ) ) ) call( "rnorm" )
```

Definition at line 63 of file Language.h.

**5.2.3.116 Language\_Impl() [4/5]**

```
Rcpp::Language_Impl (
    const Symbol & symbol ) [explicit]
```

Creates a call using the given symbol as the function name

**Parameters**

<i>symbol</i>	symbol name to call
---------------	---------------------

Language( [Symbol](#)("rnorm") ) makes a SEXP similar to this:

```
call( "rnorm" )
```

Definition at line 75 of file Language.h.



**5.2.3.117 Language\_Impl()** [5/5]

```
Rcpp::Language_Impl (
    SEXP x )
```

Attempts to convert the SEXP to a call

**Exceptions**

<i>not_compatible</i>	if the SEXP could not be converted to a call using <code>as.call</code>
-----------------------	---

Definition at line 50 of file Language.h.

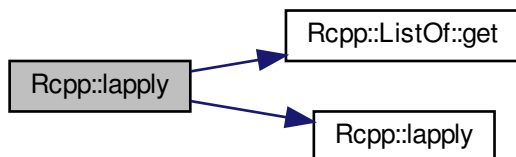
**5.2.3.118 lapply()** [1/2]

```
template<typename T , typename Function >
List Rcpp::lapply (
    const ListOf< T > & t,
    Function fun )
```

Definition at line 129 of file ListOf.h.

References `Rcpp::ListOf< T >::get()`, and `lapply()`.

Here is the call graph for this function:

**5.2.3.119 lapply()** [2/2]

```
template<int RTYPE, bool NA, typename T , typename Function >
sugar::Lapply<RTYPE,NA,T,Function> Rcpp::lapply (
    const Rcpp::VectorBase< RTYPE, NA, T > & t,
    Function fun ) [inline]
```

Definition at line 55 of file lapply.h.

Referenced by `lapply()`.

### 5.2.3.120 leaps\_thru\_end\_of()

```
static int Rcpp::leaps_thru_end_of (
    const int y ) [static]
```

Definition at line 1222 of file date.cpp.

Referenced by timesub().

### 5.2.3.121 lock()

```
void Rcpp::lock (
    bool bindings = false )
```

locks this environment. See ?lockEnvironment

#### Parameters

<i>bindings</i>	also lock the bindings of this environment ?
-----------------	--

Definition at line 264 of file Environment.h.

### 5.2.3.122 lockBinding()

```
void Rcpp::lockBinding (
    const std::string & name )
```

Locks the given binding in the environment. see ?bindingIsLocked

#### Exceptions

<i>no_such_binding</i>	if there is no such binding in this environment
------------------------	---

Definition at line 274 of file Environment.h.

References exists().

Here is the call graph for this function:



### 5.2.3.123 lower\_tri()

```
template<int RTYPE, bool NA, typename T >  
sugar::LowerTri<RTYPE, NA, T> Rcpp::lower_tri (  
    const Rcpp::MatrixBase< RTYPE, NA, T > & lhs,  
    bool diag = false ) [inline]
```

Definition at line 63 of file lower\_tri.h.

References diag().

Here is the call graph for this function:



### 5.2.3.124 ls()

```
SEXP Rcpp::ls (  
    bool all ) const
```

The list of objects in the environment

the same as calling this from R:

```
ls( envir = this, all = all )
```

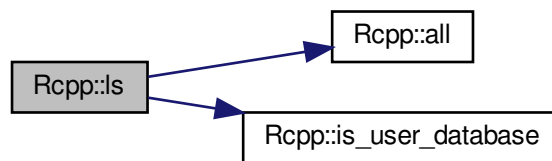
## Parameters

<i>all</i>	same meaning as in ?!s
------------	------------------------

Definition at line 85 of file Environment.h.

References `all()`, and `is_user_database()`.

Here is the call graph for this function:



### 5.2.3.125 `make_string_transformer()`

```

template<typename UnaryOperator >
StringTransformer<UnaryOperator> Rcpp::make_string_transformer (
    const UnaryOperator & fun )
  
```

Definition at line 50 of file StringTransformer.h.

### 5.2.3.126 `mapply()` [1/4]

```

template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
sugar::Mapply_2<RTYPE,NA_1,T_1,NA_2,T_2,Function> Rcpp::mapply (
    const Rcpp::VectorBase< RTYPE, NA_1, T_1 > & t1,
    const Rcpp::VectorBase< RTYPE, NA_2, T_2 > & t2,
    Function fun ) [inline]
  
```

Definition at line 126 of file mapply\_2.h.

References `Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref()`.

Here is the call graph for this function:



### 5.2.3.127 mapply() [2/4]

```

template<int RTYPE, bool NA_1, typename T_1 , typename Function >
sugar::Mapply_2_Vector_Primitive<RTYPE,NA_1,T_1,double,Function> Rcpp::mapply (
    const Rcpp::VectorBase< RTYPE, NA_1, T_1 > & t1,
    double t2,
    Function fun ) [inline]
  
```

Definition at line 132 of file mapply\_2.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Here is the call graph for this function:



### 5.2.3.128 mapply() [3/4]

```

template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_←
_3, bool NA_3, typename T_3 , typename Function >
sugar::Mapply_3<RTYPE_1,NA_1,T_1,RTYPE_2,NA_2,T_2,RTYPE_3, NA_3,T_3,Function> Rcpp::mapply (
    const Rcpp::VectorBase< RTYPE_1, NA_1, T_1 > & t1,
    const Rcpp::VectorBase< RTYPE_2, NA_2, T_2 > & t2,
    const Rcpp::VectorBase< RTYPE_3, NA_3, T_3 > & t3,
    Function fun ) [inline]
  
```

Definition at line 76 of file mapply\_3.h.

### 5.2.3.129 mapply() [4/4]

```
template<int RTYPE, bool NA_2, typename T_2 , typename Function >
sugar::Mapply_2_Primitive_Vector<RTYPE,double, NA_2,T_2,Function> Rcpp::mapply (
    double t1,
    const Rcpp::VectorBase< RTYPE, NA_2, T_2 > & t2,
    Function fun ) [inline]
```

Definition at line 138 of file mapply\_2.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Here is the call graph for this function:



### 5.2.3.130 match()

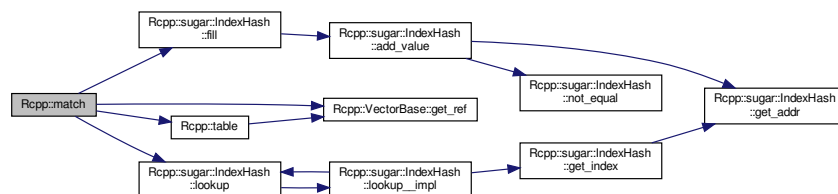
```
template<int RTYPE, bool NA, typename T , bool RHS_NA, typename RHS_T >
IntegerVector Rcpp::match (
    const VectorBase< RTYPE, NA, T > & x,
    const VectorBase< RTYPE, RHS_NA, RHS_T > & table_ ) [inline]
```

Definition at line 28 of file match.h.

References Rcpp::sugar::IndexHash< RTYPE >::fill(), Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref(), Rcpp::sugar::IndexHash< RTYPE >::lookup(), and table().

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::get\_indices().

Here is the call graph for this function:



**5.2.3.131 max()**

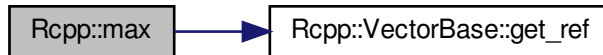
```
template<int RTYPE, bool NA, typename T >
sugar::Max<RTYPE,NA,T> Rcpp::max (
    const VectorBase< RTYPE, NA, T > & x )
```

Definition at line 82 of file max.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Referenced by getnum(), Rcpp::sugar::Max< RTYPE, NA, T >::operator STORAGE(), Rcpp::sugar::Max< RTYPE, false, T >::operator STORAGE(), operator<<(), and runif().

Here is the call graph for this function:

**5.2.3.132 mean() [1/4]**

```
template<bool NA, typename T >
sugar::Mean<CPLXSP,NA,T> Rcpp::mean (
    const VectorBase< CPLXSP, NA, T > & t ) [inline]
```

Definition at line 150 of file mean.h.

**5.2.3.133 mean() [2/4]**

```
template<bool NA, typename T >
sugar::Mean<INTSP,NA,T> Rcpp::mean (
    const VectorBase< INTSP, NA, T > & t ) [inline]
```

Definition at line 145 of file mean.h.

**5.2.3.134 mean()** [3/4]

```
template<bool NA, typename T >
sugar::Mean<LGLSXP,NA,T> Rcpp::mean (
    const VectorBase< LGLSXP, NA, T > & t ) [inline]
```

Definition at line 155 of file mean.h.

**5.2.3.135 mean()** [4/4]

```
template<bool NA, typename T >
sugar::Mean<REALSXP,NA,T> Rcpp::mean (
    const VectorBase< REALSXP, NA, T > & t ) [inline]
```

Definition at line 140 of file mean.h.

Referenced by Rcpp::sugar::Var< RTYPE, NA, T >::get(), Rcpp::sugar::Var< CPLXSXP, NA, T >::get(), and rnorm().

**5.2.3.136 median()**

```
template<int RTYPE, bool NA, typename T >
sugar::median_detail::result<RTYPE>::type Rcpp::median (
    const Rcpp::VectorBase< RTYPE, NA, T > & x,
    bool na_rm = false ) [inline]
```

Definition at line 283 of file median.h.

**5.2.3.137 message()**

```
void Rcpp::message (
    SEXP s ) [inline]
```

Definition at line 26 of file message.h.

References base\_env().

Referenced by stop(), and warning().

Here is the call graph for this function:





**5.2.3.138 min()**

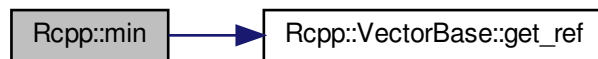
```
template<int RTYPE, bool NA, typename T >
sugar::Min<RTYPE, NA, T> Rcpp::min (
    const VectorBase< RTYPE, NA, T > & x )
```

Definition at line 82 of file min.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Referenced by Rcpp::sugar::WhichMax< RTYPE, NA, T >::get(), Rcpp::sugar::WhichMax< RTYPE, false, T >::get(), Rcpp::sugar::WhichMin< RTYPE, NA, T >::get(), Rcpp::sugar::WhichMin< RTYPE, false, T >::get(), getnum(), Rcpp::sugar::Min< RTYPE, NA, T >::operator STORAGE(), Rcpp::sugar::Min< RTYPE, false, T >::operator STORAGE(), operator<<(), and runif().

Here is the call graph for this function:

**5.2.3.139 mktime00()**

```
double Rcpp::mktime00 (
    struct tm & tm ) [inline]
```

Definition at line 130 of file routines.h.

References GET\_CALLABLE, and tm.

Referenced by Rcpp::Date::Date(), and registerFunctions().

**5.2.3.140 na\_omit()**

```
template<int RTYPE, bool NA, typename T >
Vector<RTYPE> Rcpp::na_omit (
    const VectorBase< RTYPE, NA, T > & t ) [inline]
```

Definition at line 75 of file na\_omit.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Here is the call graph for this function:

**5.2.3.141 Named() [1/2]**

```
Argument Rcpp::Named (
    const std::string & name ) [inline]
```

Definition at line 40 of file Named.h.

Referenced by Rcpp::attributes::createDirectory(), fastLm(), and Rcpp::attributes::showWarning().

**5.2.3.142 Named() [2/2]**

```
template<typename T >
traits::named_object<T> Rcpp::Named (
    const std::string & name,
    const T & o ) [inline]
```

Definition at line 44 of file Named.h.

**5.2.3.143 namespace\_env()**

```
static Environment_Impl Rcpp::namespace_env (
    const std::string & package ) [static]
```

## Parameters

<i>name</i>	the name of the package of which we want the namespace
-------------	--

## Returns

the namespace of the package

## Exceptions

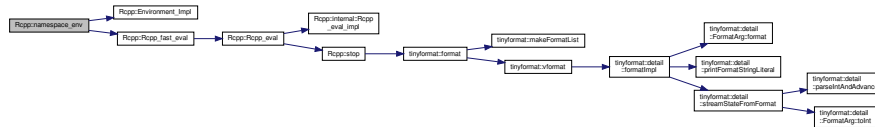
<i>no_such_namespace</i>
--------------------------

Definition at line 371 of file Environment.h.

References Environment\_Impl(), and Rcpp\_fast\_eval().

Referenced by Rcpp::attributes::checkRSignature(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

Here is the call graph for this function:



## 5.2.3.144 new\_child()

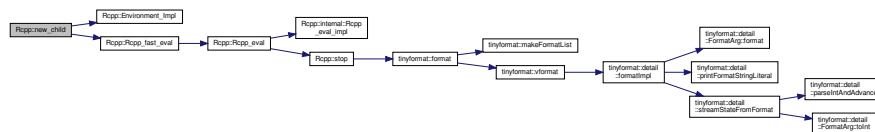
```
Environment_Impl Rcpp::new_child (
    bool hashed ) const
```

creates a new environment whose this is the parent

Definition at line 394 of file Environment.h.

References Environment\_Impl(), and Rcpp\_fast\_eval().

Here is the call graph for this function:



### 5.2.3.145 `new_env()` [1/2]

```
Environment Rcpp::new_env (  
    int size = 29 ) [inline]
```

Definition at line 47 of file Environment.h.

References [\\_](#).

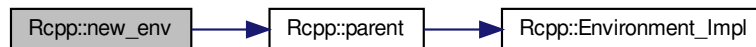
### 5.2.3.146 `new_env()` [2/2]

```
Environment Rcpp::new_env (  
    SEXP parent,  
    int size = 29 ) [inline]
```

Definition at line 52 of file Environment.h.

References [\\_](#), and [parent\(\)](#).

Here is the call graph for this function:



### 5.2.3.147 `no_init()` [1/2]

```
no_init_matrix Rcpp::no_init (  
    int nr,  
    int nc ) [inline]
```

Definition at line 81 of file no\_init.h.

**5.2.3.148 no\_init()** [2/2]

```
no_init_vector Rcpp::no_init (
    R_xlen_t size ) [inline]
```

Definition at line 77 of file no\_init.h.

Referenced by `Rcpp::sugar::EmpiricalSample()`, `Rcpp::sugar::IndexHash< RTYPE >::fill_and_get_duplicated()`, `Rcpp::sugar::SelfHash< RTYPE >::fill_and_self_match()`, `Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get()`, `Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get()`, `Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get()`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_vec()`, `Rcpp::sugar::IndexHash< RTYPE >::keys()`, `Rcpp::sugar::na_omit_impl()`, `Rcpp::sugar::Table< RTYPE, TABLE_T >::operator IntegerVector()`, `Rcpp::sugar::SampleNoReplace()`, `Rcpp::sugar::SampleReplace()`, `trimws()`, and `Rcpp::sugar::WalkerSample()`.

**5.2.3.149 noNA()** [1/3]

```
template<int RTYPE, bool NA, typename VECTOR >
sugar::Nona<RTYPE,NA,VECTOR> Rcpp::noNA (
    const Rcpp::VectorBase< RTYPE, NA, VECTOR > & vec ) [inline]
```

Definition at line 74 of file nona.h.

Referenced by `convolve11cpp()`.

**5.2.3.150 noNA()** [2/3]

```
sugar::NonaPrimitive<double> Rcpp::noNA (
    double x ) [inline]
```

Definition at line 78 of file nona.h.

**5.2.3.151 noNA()** [3/3]

```
sugar::NonaPrimitive<int> Rcpp::noNA (
    int x ) [inline]
```

Definition at line 81 of file nona.h.

### 5.2.3.152 operator SEXP()

```
Rcpp::operator SEXP ( ) const [inline]
```

Definition at line 53 of file StretchyList.h.

### 5.2.3.153 operator"!="() [1/4]

```
bool Rcpp::operator!= (
    const Date & d1,
    const Date & d2 ) [inline]
```

Definition at line 170 of file Date.h.

### 5.2.3.154 operator"!="() [2/4]

```
bool Rcpp::operator!= (
    const Datetime & d1,
    const Datetime & d2 ) [inline]
```

Definition at line 164 of file Datetime.h.

### 5.2.3.155 operator"!="() [3/4]

```
bool Rcpp::operator!= (
    const String::const_StringProxy & lhs,
    const String & rhs ) [inline]
```

Definition at line 731 of file String.h.

### 5.2.3.156 operator"!="() [4/4]

```
bool Rcpp::operator!= (
    const String::StringProxy & lhs,
    const String & rhs ) [inline]
```

Definition at line 723 of file String.h.

**5.2.3.157 operator\*() [1/5]**

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Times_Vector_Primitive_nona< RTYPE ,
NA , T > >::type Rcpp::operator* (
    const typename sugar::NonaPrimitive< U > & rhs,
    const VectorBase< RTYPE, NA, T > & lhs ) [inline]
```

Definition at line 459 of file times.h.

**5.2.3.158 operator\*() [2/5]**

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Times_Vector_Primitive< RTYPE , NA ,
T > >::type Rcpp::operator* (
    const U & rhs,
    const VectorBase< RTYPE, NA, T > & lhs ) [inline]
```

Definition at line 439 of file times.h.

**5.2.3.159 operator\*() [3/5]**

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
sugar::Times_Vector_Vector< RTYPE , LHS_NA, LHS_T, RHS_NA, RHS_T > Rcpp::operator* (
    const VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 473 of file times.h.

**5.2.3.160 operator\*() [4/5]**

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Times_Vector_Primitive_nona<RTYPE,NA,T>
>::type Rcpp::operator* (
    const VectorBase< RTYPE, NA, T > & lhs,
    const typename sugar::NonaPrimitive< U > & rhs ) [inline]
```

Definition at line 450 of file times.h.

### 5.2.3.161 operator\*() [5/5]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Times_Vector_Primitive<RTYPE,NA,T>
>::type Rcpp::operator* (
    const VectorBase< RTYPE, NA, T > & lhs,
    const U & rhs ) [inline]
```

Definition at line 429 of file times.h.

### 5.2.3.162 operator+() [1/8]

```
Date Rcpp::operator+ (
    const Date & date,
    int offset ) [inline]
```

Definition at line 156 of file Date.h.

### 5.2.3.163 operator+() [2/8]

```
Datetime Rcpp::operator+ (
    const Datetime & datetime,
    double offset ) [inline]
```

Definition at line 138 of file Datetime.h.

### 5.2.3.164 operator+() [3/8]

```
Datetime Rcpp::operator+ (
    const Datetime & datetime,
    int offset ) [inline]
```

Definition at line 148 of file Datetime.h.



**5.2.3.165 operator+()** [4/8]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Plus_Vector_Primitive_nona< RTYPE ,
NA , T> >::type Rcpp::operator+ (
    const typename sugar::NonaPrimitive< U > & rhs,
    const VectorBase< RTYPE, NA, T > & lhs ) [inline]
```

Definition at line 478 of file plus.h.

**5.2.3.166 operator+()** [5/8]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, typename sugar::Plus_Vector_Primitive< RTYPE
, NA , T> >::type Rcpp::operator+ (
    const U & rhs,
    const VectorBase< RTYPE, NA, T > & lhs ) [inline]
```

Definition at line 458 of file plus.h.

**5.2.3.167 operator+()** [6/8]

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
sugar::Plus_Vector_Vector< RTYPE , LHS_NA, LHS_T, RHS_NA, RHS_T > Rcpp::operator+ (
    const VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 492 of file plus.h.

**5.2.3.168 operator+()** [7/8]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Plus_Vector_Primitive_nona<RTYPE,NA,T>
>::type Rcpp::operator+ (
    const VectorBase< RTYPE, NA, T > & lhs,
    const typename sugar::NonaPrimitive< U > & rhs ) [inline]
```

Definition at line 469 of file plus.h.

### 5.2.3.169 operator+() [8/8]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, typename sugar::Plus_Vector_Primitive<RTYPE,NA,T>
>::type Rcpp::operator+ (
    const VectorBase< RTYPE, NA, T > & lhs,
    const U & rhs ) [inline]
```

Definition at line 448 of file plus.h.

### 5.2.3.170 operator-() [1/5]

```
double Rcpp::operator- (
    const Date & d1,
    const Date & d2 ) [inline]
```

Definition at line 164 of file Date.h.

### 5.2.3.171 operator-() [2/5]

```
double Rcpp::operator- (
    const Datetime & d1,
    const Datetime & d2 ) [inline]
```

Definition at line 158 of file Datetime.h.

### 5.2.3.172 operator-() [3/5]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Minus_Primitive_Vector< RTYPE , NA,T>
>::type Rcpp::operator- (
    const U & lhs,
    const VectorBase< RTYPE, NA, T > & rhs ) [inline]
```

Definition at line 427 of file minus.h.

**5.2.3.173 operator-() [4/5]**

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
sugar::Minus_Vector_Vector< RTYPE , LHS_NA, LHS_T, RHS_NA, RHS_T > Rcpp::operator- (
    const VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 440 of file minus.h.

**5.2.3.174 operator-() [5/5]**

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Minus_Vector_Primitive< RTYPE , NA, T
> >::type Rcpp::operator- (
    const VectorBase< RTYPE, NA, T > & lhs,
    const U & rhs ) [inline]
```

Definition at line 417 of file minus.h.

**5.2.3.175 operator/() [1/3]**

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Divides_Primitive_Vector< RTYPE ,
NA,T> >::type Rcpp::operator/ (
    const U & lhs,
    const VectorBase< RTYPE, NA, T > & rhs ) [inline]
```

Definition at line 423 of file divides.h.

**5.2.3.176 operator/() [2/3]**

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
sugar::Divides_Vector_Vector< RTYPE , LHS_NA, LHS_T, RHS_NA, RHS_T > Rcpp::operator/ (
    const VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 436 of file divides.h.

### 5.2.3.177 operator/() [3/3]

```
template<int RTYPE, bool NA, typename T , typename U >
traits::enable_if<traits::is_convertible<typename traits::remove_const_and_reference<U>::type,
typename traits::storage_type<RTYPE>::type>::value, sugar::Divides_Vector_Primitive< RTYPE , NA,
T > >::type Rcpp::operator/ (
    const VectorBase< RTYPE, NA, T > & lhs,
    const U & rhs ) [inline]
```

Definition at line 413 of file divides.h.

### 5.2.3.178 operator<() [1/2]

```
bool Rcpp::operator< (
    const Date & d1,
    const Date & d2 ) [inline]
```

Definition at line 165 of file Date.h.

### 5.2.3.179 operator<() [2/2]

```
bool Rcpp::operator< (
    const Datetime & d1,
    const Datetime & d2 ) [inline]
```

Definition at line 159 of file Datetime.h.

### 5.2.3.180 operator<<() [1/10]

```
std::ostream& Rcpp::operator<< (
    std::ostream & os,
    const Date d ) [inline]
```

Definition at line 172 of file Date.h.

**5.2.3.181 operator<<() [2/10]**

```
std::ostream& Rcpp::operator<< (
    std::ostream & os,
    const Datetime d ) [inline]
```

Definition at line 166 of file Datetime.h.

**5.2.3.182 operator<<() [3/10]**

```
std::ostream& Rcpp::operator<< (
    std::ostream & os,
    const newDatetimeVector d ) [inline]
```

Definition at line 81 of file newDatetimeVector.h.

**5.2.3.183 operator<<() [4/10]**

```
std::ostream& Rcpp::operator<< (
    std::ostream & os,
    const newDateVector d ) [inline]
```

Definition at line 65 of file newDateVector.h.

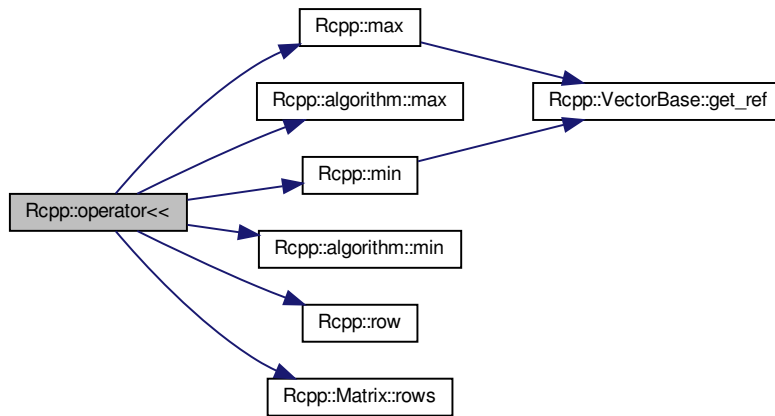
**5.2.3.184 operator<<() [5/10]**

```
template<template< class > class StoragePolicy>
std::ostream& Rcpp::operator<< (
    std::ostream & s,
    const Matrix< INTSXP, StoragePolicy > & rhs ) [inline]
```

Definition at line 286 of file Matrix.h.

References `max()`, `Rcpp::algorithm::max()`, `min()`, `Rcpp::algorithm::min()`, `row()`, and `Rcpp::Matrix< RTYPE, StoragePolicy >::rows()`.

Here is the call graph for this function:



### 5.2.3.185 `operator<<()` [6/10]

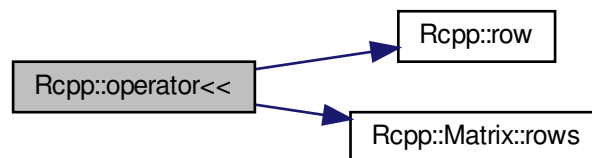
```

template<template< class > class StoragePolicy>
std::ostream& Rcpp::operator<< (
    std::ostream & s,
    const Matrix< REALSXP, StoragePolicy > & rhs ) [inline]
  
```

Definition at line 218 of file Matrix.h.

References `row()`, and `Rcpp::Matrix< RTYPE, StoragePolicy >::rows()`.

Here is the call graph for this function:



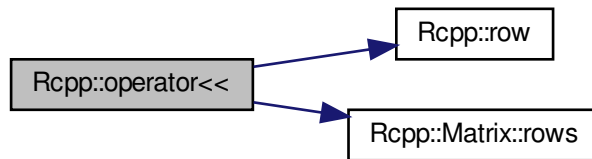
**5.2.3.186 operator<<()** [7/10]

```
template<int RTYPE, template< class > class StoragePolicy>
std::ostream& Rcpp::operator<< (
    std::ostream & s,
    const Matrix< RTYPE, StoragePolicy > & rhs ) [inline]
```

Definition at line 379 of file Matrix.h.

References [row\(\)](#), and [Rcpp::Matrix< RTYPE, StoragePolicy >::rows\(\)](#).

Here is the call graph for this function:

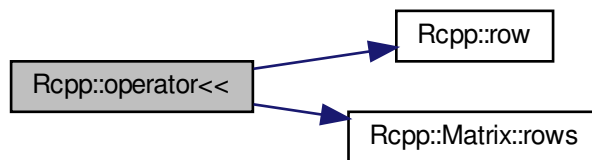
**5.2.3.187 operator<<()** [8/10]

```
template<template< class > class StoragePolicy>
std::ostream& Rcpp::operator<< (
    std::ostream & s,
    const Matrix< STRSXP, StoragePolicy > & rhs ) [inline]
```

Definition at line 352 of file Matrix.h.

References [row\(\)](#), and [Rcpp::Matrix< RTYPE, StoragePolicy >::rows\(\)](#).

Here is the call graph for this function:



**5.2.3.188 operator<<() [9/10]**

```
template<int RTYPE, template< class > class StoragePolicy>
std::ostream& Rcpp::operator<< (
    std::ostream & s,
    const Vector< RTYPE, StoragePolicy > & rhs ) [inline]
```

Definition at line 1142 of file Vector.h.

**5.2.3.189 operator<<() [10/10]**

```
template<template< class > class StoragePolicy>
std::ostream& Rcpp::operator<< (
    std::ostream & s,
    const Vector< STRSXP, StoragePolicy > & rhs ) [inline]
```

Definition at line 1161 of file Vector.h.

**5.2.3.190 operator<=() [1/2]**

```
bool Rcpp::operator<= (
    const Date & d1,
    const Date & d2 ) [inline]
```

Definition at line 169 of file Date.h.

**5.2.3.191 operator<=() [2/2]**

```
bool Rcpp::operator<= (
    const Datetime & d1,
    const Datetime & d2 ) [inline]
```

Definition at line 163 of file Datetime.h.



**5.2.3.192 operator==( ) [1/4]**

```
bool Rcpp::operator==(
    const Date & d1,
    const Date & d2 ) [inline]
```

Definition at line 167 of file Date.h.

Referenced by RCPP\_API\_CLASS().

**5.2.3.193 operator==( ) [2/4]**

```
bool Rcpp::operator==(
    const Datetime & d1,
    const Datetime & d2 ) [inline]
```

Definition at line 161 of file Datetime.h.

**5.2.3.194 operator==( ) [3/4]**

```
bool Rcpp::operator==(
    const String::const_StringProxy & lhs,
    const String & rhs ) [inline]
```

Definition at line 727 of file String.h.

**5.2.3.195 operator==( ) [4/4]**

```
bool Rcpp::operator==(
    const String::StringProxy & lhs,
    const String & rhs ) [inline]
```

Definition at line 719 of file String.h.

**5.2.3.196 operator>( ) [1/2]**

```
bool Rcpp::operator>(
    const Date & d1,
    const Date & d2 ) [inline]
```

Definition at line 166 of file Date.h.

### 5.2.3.197 operator>() [2/2]

```
bool Rcpp::operator> (
    const Datetime & d1,
    const Datetime & d2 ) [inline]
```

Definition at line 160 of file Datetime.h.

### 5.2.3.198 operator>=() [1/2]

```
bool Rcpp::operator>= (
    const Date & d1,
    const Date & d2 ) [inline]
```

Definition at line 168 of file Date.h.

### 5.2.3.199 operator>=() [2/2]

```
bool Rcpp::operator>= (
    const Datetime & d1,
    const Datetime & d2 ) [inline]
```

Definition at line 162 of file Datetime.h.

### 5.2.3.200 outer()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
sugar::Outer<RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function> Rcpp::outer (
    const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs,
    Function fun ) [inline]
```

Definition at line 80 of file outer.h.

### 5.2.3.201 pairlist()

```
SEXP Rcpp::pairlist ( ) [inline]
```

Definition at line 30 of file grow.h.

Referenced by Rcpp::DottedPairImpl< CLASS >::replace().

### 5.2.3.202 Pairlist\_Impl() [1/2]

```
Rcpp::Pairlist_Impl ( )
```

Definition at line 41 of file Pairlist.h.

### 5.2.3.203 Pairlist\_Impl() [2/2]

```
Rcpp::Pairlist_Impl (
    SEXP x )
```

Definition at line 42 of file Pairlist.h.

### 5.2.3.204 parent()

```
Environment_Impl Rcpp::parent ( ) const
```

The parent environment of this environment

Definition at line 387 of file Environment.h.

References Environment\_Impl().

Referenced by Rcpp::ConstMatrixColumn< RTYPE >::ConstMatrixColumn(), class\_< Class >::derives(), Rcpp::← MatrixColumn< RTYPE >::MatrixColumn(), and new\_env().

Here is the call graph for this function:



### 5.2.3.205 pexp()

```
template<bool NA, typename T >
stats::P1<REALSXP,NA,T> Rcpp::pexp (
    const Rcpp::VectorBase< REALSXP, NA, T > & x,
    double shape,
    bool lower = true,
    bool log = false ) [inline]
```

Definition at line 86 of file exp.h.

References Rcpp::algorithm::log().

Here is the call graph for this function:



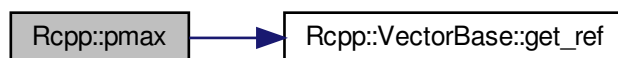
### 5.2.3.206 pmax() [1/3]

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
sugar::Pmax_Vector_Vector<RTYPE,LHS_NA,LHS_T,RHS_NA,RHS_T> Rcpp::pmax (
    const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 158 of file pmax.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Here is the call graph for this function:



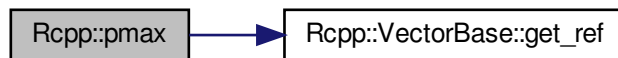
**5.2.3.207 pmax()** [2/3]

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
sugar::Pmax_Vector_Primitive<RTYPE,LHS_NA,LHS_T> Rcpp::pmax (
    const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    typename Rcpp::traits::storage_type< RTYPE >::type rhs ) [inline]
```

Definition at line 170 of file pmax.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Here is the call graph for this function:

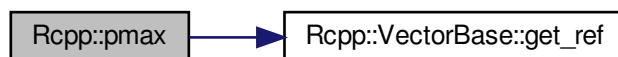
**5.2.3.208 pmax()** [3/3]

```
template<int RTYPE, bool RHS_NA, typename RHS_T >
sugar::Pmax_Vector_Primitive<RTYPE,RHS_NA,RHS_T> Rcpp::pmax (
    typename Rcpp::traits::storage_type< RTYPE >::type lhs,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 183 of file pmax.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Here is the call graph for this function:



**5.2.3.209 pmin()** [1/3]

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
sugar::Pmin_Vector_Vector<RTYPE,LHS_NA,LHS_T,RHS_NA,RHS_T> Rcpp::pmin (
    const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 156 of file pmin.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Here is the call graph for this function:

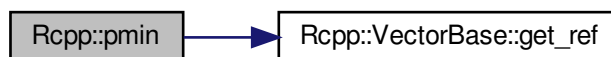
**5.2.3.210 pmin()** [2/3]

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
sugar::Pmin_Vector_Primitive<RTYPE,LHS_NA,LHS_T> Rcpp::pmin (
    const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    typename Rcpp::traits::storage_type< RTYPE >::type rhs ) [inline]
```

Definition at line 168 of file pmin.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Here is the call graph for this function:



**5.2.3.211 pmin()** [3/3]

```
template<int RTYPE, bool RHS_NA, typename RHS_T >
sugar::Pmin_Vector_Primitive<RTYPE,RHS_NA,RHS_T> Rcpp::pmin (
    typename Rcpp::traits::storage_type< RTYPE >::type lhs,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 181 of file pmin.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Here is the call graph for this function:

**5.2.3.212 pow()**

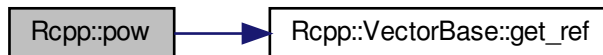
```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >
sugar::Pow<RTYPE,NA,T,EXPONENT_TYPE> Rcpp::pow (
    const VectorBase< RTYPE, NA, T > & t,
    EXPONENT_TYPE exponent ) [inline]
```

Definition at line 79 of file pow.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Referenced by Rcpp::stats::dweibull\_1(), Rcpp::sugar::Var< RTYPE, NA, T >::get(), Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT\_TYPE >::operator[](), Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT\_TYPE >::operator[](), Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT\_TYPE >::operator[](), Rcpp::stats::pweibull\_1(), and Rcpp::stats::qweibull\_1().

Here is the call graph for this function:



### 5.2.3.213 print()

```
void Rcpp::print (
    SEXP s ) [inline]
```

Definition at line 25 of file print.h.

### 5.2.3.214 push\_back()

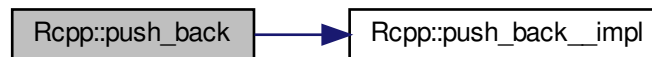
```
template<typename T >
StretchyList_Impl& Rcpp::push_back (
    const T & obj ) [inline]
```

Definition at line 58 of file StretchyList.h.

References `push_back__impl()`.

Referenced by `class_< Class >::AddMethod()`.

Here is the call graph for this function:



### 5.2.3.215 push\_back\_\_impl() [1/2]

```
template<typename T >
StretchyList_Impl& Rcpp::push_back__impl (
    const T & obj,
    traits::false_type ) [private]
```



### 5.2.3.216 `push_back__impl()` [2/2]

```
template<typename T >
StretchyList_Impl& Rcpp::push_back__impl (
    const T & obj,
    traits::true_type ) [private]
```

Referenced by `push_back()`.

### 5.2.3.217 `push_front()`

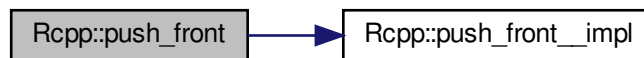
```
template<typename T >
StretchyList_Impl& Rcpp::push_front (
    const T & obj ) [inline]
```

Definition at line 63 of file `StretchyList.h`.

References `push_front__impl()`.

Referenced by `Rcpp::DottedPairImpl< CLASS >::insert()`.

Here is the call graph for this function:



### 5.2.3.218 `push_front__impl()` [1/2]

```
template<typename T >
StretchyList_Impl& Rcpp::push_front__impl (
    const T & obj,
    traits::false_type ) [private]
```

**5.2.3.219 push\_front\_\_impl() [2/2]**

```
template<typename T >
StretchyList_Impl& Rcpp::push_front__impl (
    const T & obj,
    traits::true_type ) [private]
```

Referenced by push\_front().

**5.2.3.220 qexp()**

```
template<bool NA, typename T >
stats::Q1<REALSXP,NA,T> Rcpp::qexp (
    const Rcpp::VectorBase< REALSXP, NA, T > & x,
    double shape,
    bool lower = true,
    bool log = false ) [inline]
```

Definition at line 91 of file exp.h.

References Rcpp::algorithm::log().

Here is the call graph for this function:

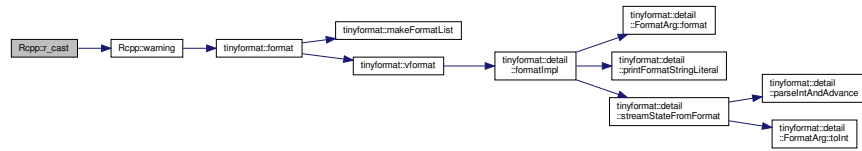
**5.2.3.221 r\_cast()**

```
template<int TARGET>
SEXP Rcpp::r_cast (
    SEXP x )
```

Definition at line 164 of file r\_cast.h.

References warning().

Here is the call graph for this function:



### 5.2.3.222 range()

```

template<int RTYPE, bool NA, typename T >
sugar::Range<RTYPE,NA,T> Rcpp::range (
    const VectorBase< RTYPE, NA, T > & x )
  
```

Definition at line 86 of file range.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::operator[ ]().

Here is the call graph for this function:



### 5.2.3.223 rbeta()

```

NumericVector Rcpp::rbeta (
    int n,
    double a,
    double b ) [inline]
  
```

Definition at line 100 of file random.h.

**5.2.3.224 rbinom()**

```
NumericVector Rcpp::rbinom (  
    int n,  
    double pin,  
    double pp ) [inline]
```

Definition at line 104 of file random.h.

**5.2.3.225 rcauchy() [1/3]**

```
NumericVector Rcpp::rcauchy (  
    int n ) [inline]
```

Definition at line 128 of file random.h.

**5.2.3.226 rcauchy() [2/3]**

```
NumericVector Rcpp::rcauchy (  
    int n,  
    double location ) [inline]
```

Definition at line 118 of file random.h.

**5.2.3.227 rcauchy() [3/3]**

```
NumericVector Rcpp::rcauchy (  
    int n,  
    double location,  
    double scale ) [inline]
```

Definition at line 108 of file random.h.

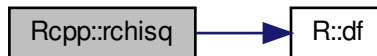
### 5.2.3.228 rchisq()

```
NumericVector Rcpp::rchisq (  
    int n,  
    double df ) [inline]
```

Definition at line 132 of file random.h.

References R::df().

Here is the call graph for this function:



### 5.2.3.229 RCPP\_API\_CLASS() [1/14]

```
Rcpp::RCPP_API_CLASS (  
    DottedPair_Impl )
```

### 5.2.3.230 RCPP\_API\_CLASS() [2/14]

```
Rcpp::RCPP_API_CLASS (  
    Environment_Impl )
```

### 5.2.3.231 RCPP\_API\_CLASS() [3/14]

```
Rcpp::RCPP_API_CLASS (  
    Formula_Impl )
```

### 5.2.3.232 RCPP\_API\_CLASS() [4/14]

```
Rcpp::RCPP_API_CLASS (  
    Function_Impl )
```

functions Finds a function. By default, searches from the global environment

## Parameters

<i>name</i>	name of the function
<i>env</i>	an environment where to search the function
<i>ns</i>	name of the namespace in which to search the function

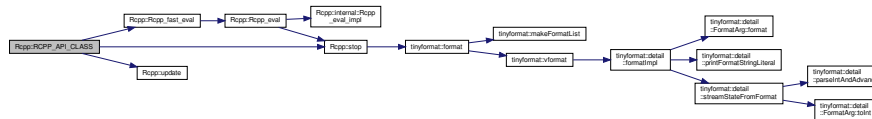
Returns the environment of this function

Returns the body of the function

Definition at line 34 of file Function.h.

References `Rcpp_fast_eval()`, `RCPPE_GENERATE_CTOR_ASSIGN`, `Rcpp_icons`, `stop()`, and `update()`.

Here is the call graph for this function:



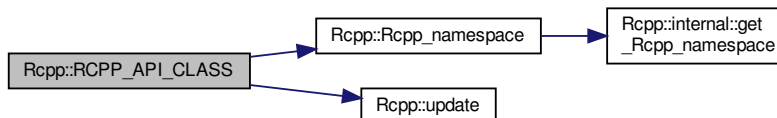
### 5.2.3.233 RCPP\_API\_CLASS() [5/14]

```
Rcpp::RCPP_API_CLASS (
    InternalFunction_Impl )
```

Definition at line 35 of file InternalFunction.h.

References `RCPPE_GENERATE_CTOR_ASSIGN`, `Rcpp_namespace()`, and `update()`.

Here is the call graph for this function:



**5.2.3.234 RCPP\_API\_CLASS()** [6/14]

```
Rcpp::RCPP_API_CLASS (
    Language_Impl )
```

C++ wrapper around calls (LANGSXP SEXP)

This represents calls that can be evaluated

**5.2.3.235 RCPP\_API\_CLASS()** [7/14]

```
Rcpp::RCPP_API_CLASS (
    Pairlist_Impl )
```

**5.2.3.236 RCPP\_API\_CLASS()** [8/14]

```
Rcpp::RCPP_API_CLASS (
    Promise_Impl )
```

Return the result of the PRSEEN macro

Return the result of the PRVALUE macro on the promise

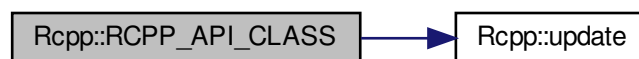
The promise expression: PRCODE

The promise environment : PRENV

Definition at line 27 of file Promise.h.

References RCPP\_GENERATE\_CTOR\_ASSIGN, and update().

Here is the call graph for this function:



**5.2.3.237 RCPP\_API\_CLASS()** [9/14]

```
Rcpp::RCPP_API_CLASS (
    Reference_Impl )
```

S4 object (of a reference class)

**5.2.3.238 RCPP\_API\_CLASS()** [10/14]

```
Rcpp::RCPP_API_CLASS (
    RObject_Impl )
```

default constructor. uses R\_NilValue

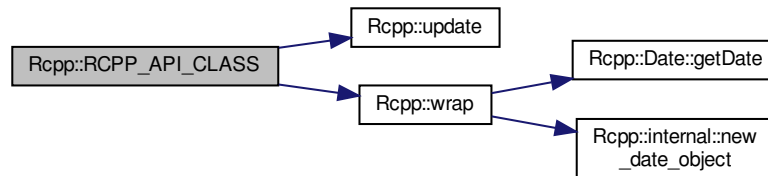
wraps a SEXP. The SEXP is automatically protected from garbage collection by this object and the protection vanishes when this object is destroyed

Assignment operator. Set this SEXP to the given SEXP

Definition at line 27 of file RObject.h.

References RCPP\_GENERATE\_CTOR\_ASSIGN, update(), and wrap().

Here is the call graph for this function:

**5.2.3.239 RCPP\_API\_CLASS()** [11/14]

```
Rcpp::RCPP_API_CLASS (
    S4_Impl )
```

S4 object checks that x is an S4 object and wrap it.



## Parameters

<i>x</i>	must be an S4 object
----------	----------------------

Creates an S4 object of the requested class.

## Parameters

<i>class</i>	name of the target S4 class
--------------	-----------------------------

## Exceptions

<i>S4_creation_error</i>	if class does not map to a known S4 class
--------------------------	---

Indicates if this object is an instance of the given S4 class

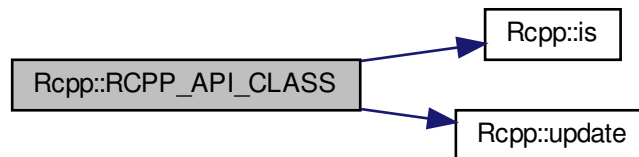
## Exceptions

<i>not_s4</i>	if x is not an S4 class
---------------	-------------------------

Definition at line 30 of file S4.h.

References `is()`, `RCPP_GENERATE_CTOR_ASSIGN`, and `update()`.

Here is the call graph for this function:



### 5.2.3.240 RCPP\_API\_CLASS() [12/14]

```
Rcpp::RCPP_API_CLASS (
    StretchyList_Impl )
```

StretchyList uses a special pairlist to provide efficient insertion at the front and the end of a pairlist.

This is a C++ abstraction of the functions `NewList`, `GrowList` and `Insert` that are found in places where a pair list has to grow efficiently, e.g. in the `R` parser (`gram.y`)

**5.2.3.241 RCPP\_API\_CLASS()** [13/14]

```
Rcpp::RCPP_API_CLASS (
    Symbol_Impl )
```

wraps the SEXP into a Symbol object.

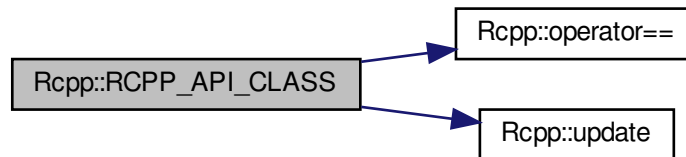
**Parameters**

x	Accepted SEXP types are SYMSXP, CHARSXP and STRSXP in the last case, the first element of the character vector is silently used
---	---

Definition at line 27 of file Symbol.h.

References operator==(), RCPP\_GENERATE\_CTOR\_ASSIGN, and update().

Here is the call graph for this function:

**5.2.3.242 RCPP\_API\_CLASS()** [14/14]

```
Rcpp::RCPP_API_CLASS (
    WeakReference_Impl )
```

wraps a weak reference

**Parameters**

x	presumably a SEXP of SEXTYPE WEAKREFSXP
---	---

**Exceptions**

<i>not_compatible</i>	if x is not a weak reference
-----------------------	------------------------------

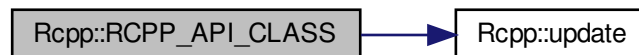
Retrieve the key

Retrieve the value

Definition at line 27 of file WeakReference.h.

References RCPP\_GENERATE\_CTOR\_ASSIGN, and update().

Here is the call graph for this function:



#### 5.2.3.243 Rcpp\_cerr\_get()

```
Rostream< false > & Rcpp::Rcpp_cerr_get ( ) [inline]
```

Definition at line 168 of file routines.h.

References GET\_CALLABLE.

Referenced by registerFunctions().

#### 5.2.3.244 Rcpp\_cout\_get()

```
Rostream< true > & Rcpp::Rcpp_cout_get ( ) [inline]
```

Definition at line 163 of file routines.h.

References GET\_CALLABLE.

Referenced by registerFunctions().

### 5.2.3.245 Rcpp\_eval()

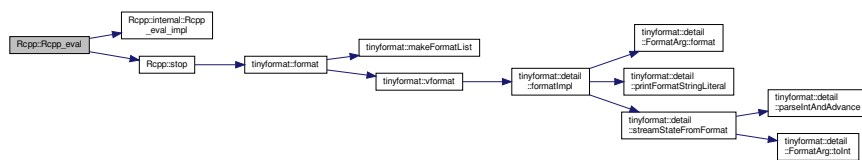
```
SEXP Rcpp::Rcpp_eval (
    SEXP expr,
    SEXP env ) [inline]
```

Definition at line 75 of file Rcpp\_eval.h.

References Rcpp::internal::Rcpp\_eval\_impl(), and stop().

Referenced by Rcpp\_fast\_eval().

Here is the call graph for this function:



### 5.2.3.246 RCPP\_EXCEPTION\_CLASS()

```
No such field Rcpp::RCPP_EXCEPTION_CLASS (
    reference_creation_error ,
    "Error creating object of reference class" )
```

### 5.2.3.247 Rcpp\_fast\_eval()

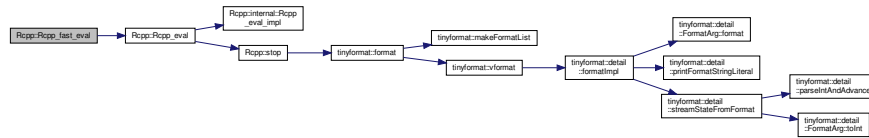
```
SEXP Rcpp::Rcpp_fast_eval (
    SEXP expr,
    SEXP env ) [inline]
```

Definition at line 68 of file Rcpp\_eval.h.

References Rcpp\_eval().

Referenced by Rcpp::internal::convert\_using\_rfunction(), eval(), Rcpp::Vector< RTYPE, StoragePolicy >::eval(), Rcpp::DataFrame\_Impl< StoragePolicy >::from\_list(), Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get(), Rcpp::FieldProxyPolicy< CLASS >::const\_FieldProxy::get(), get\_last\_call(), namespace\_env(), new\_child(), Rcpp::internal::r\_true\_cast< STRSXP >(), RCPP\_API\_CLASS(), remove(), Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::set(), and Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::set().

Here is the call graph for this function:



### 5.2.3.248 Rcpp\_lang10()

```

SEXP Rcpp::Rcpp_lang10 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8,
    SEXP x9 ) [inline]
  
```

Definition at line 127 of file lang.h.

References [Rcpp\\_list9\(\)](#).

Here is the call graph for this function:



### 5.2.3.249 Rcpp\_lang11()

```

SEXP Rcpp::Rcpp_lang11 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
  
```

```

SEXP x7,
SEXP x8,
SEXP x9,
SEXP x10 ) [inline]

```

Definition at line 143 of file lang.h.

References [Rcpp\\_list10\(\)](#).

Here is the call graph for this function:



### 5.2.3.250 Rcpp\_lang12()

```

SEXP Rcpp::Rcpp_lang12 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8,
    SEXP x9,
    SEXP x10,
    SEXP x11 ) [inline]

```

Definition at line 159 of file lang.h.

References [Rcpp\\_list11\(\)](#).

Here is the call graph for this function:



**5.2.3.251 Rcpp\_lang13()**

```
SEXP Rcpp::Rcpp_lang13 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8,
    SEXP x9,
    SEXP x10,
    SEXP x11,
    SEXP x12 ) [inline]
```

Definition at line 175 of file lang.h.

References [Rcpp\\_list12\(\)](#).

Here is the call graph for this function:

**5.2.3.252 Rcpp\_lang14()**

```
SEXP Rcpp::Rcpp_lang14 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8,
    SEXP x9,
    SEXP x10,
    SEXP x11,
    SEXP x12,
    SEXP x13 ) [inline]
```

Definition at line 191 of file lang.h.

References [Rcpp\\_list13\(\)](#).

Here is the call graph for this function:



### 5.2.3.253 Rcpp\_lang15()

```
SEXP Rcpp::Rcpp_lang15 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8,
    SEXP x9,
    SEXP x10,
    SEXP x11,
    SEXP x12,
    SEXP x13,
    SEXP x14 ) [inline]
```

Definition at line 207 of file lang.h.

References [Rcpp\\_list14\(\)](#).

Here is the call graph for this function:



### 5.2.3.254 Rcpp\_lang16()

```
SEXP Rcpp::Rcpp_lang16 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8,
    SEXP x9,
    SEXP x10,
    SEXP x11,
    SEXP x12,
    SEXP x13,
    SEXP x14,
    SEXP x15 ) [inline]
```



Definition at line 223 of file lang.h.

References `Rcpp_list15()`.

Here is the call graph for this function:



### 5.2.3.255 `Rcpp_lang17()`

```
SEXP Rcpp::Rcpp_lang17 (  
    SEXP x0,  
    SEXP x1,  
    SEXP x2,  
    SEXP x3,  
    SEXP x4,  
    SEXP x5,  
    SEXP x6,  
    SEXP x7,  
    SEXP x8,  
    SEXP x9,  
    SEXP x10,  
    SEXP x11,  
    SEXP x12,  
    SEXP x13,  
    SEXP x14,  
    SEXP x15,  
    SEXP x16 ) [inline]
```

Definition at line 239 of file lang.h.

References `Rcpp_list16()`.

Here is the call graph for this function:



### 5.2.3.256 Rcpp\_lang18()

```
SEXP Rcpp::Rcpp_lang18 (  
    SEXP x0,  
    SEXP x1,  
    SEXP x2,  
    SEXP x3,  
    SEXP x4,  
    SEXP x5,  
    SEXP x6,  
    SEXP x7,  
    SEXP x8,  
    SEXP x9,  
    SEXP x10,  
    SEXP x11,  
    SEXP x12,  
    SEXP x13,  
    SEXP x14,  
    SEXP x15,  
    SEXP x16,  
    SEXP x17 ) [inline]
```

Definition at line 255 of file lang.h.

References [Rcpp\\_list17\(\)](#).

Here is the call graph for this function:



### 5.2.3.257 Rcpp\_lang19()

```
SEXP Rcpp::Rcpp_lang19 (  
    SEXP x0,  
    SEXP x1,  
    SEXP x2,  
    SEXP x3,  
    SEXP x4,  
    SEXP x5,  
    SEXP x6,  
    SEXP x7,  
    SEXP x8,  
    SEXP x9,  
    SEXP x10,  
    SEXP x11,  
    SEXP x12,  
    SEXP x13,  
    SEXP x14,
```

```

SEXP x15,
SEXP x16,
SEXP x17,
SEXP x18 ) [inline]

```

Definition at line 271 of file lang.h.

References [Rcpp\\_list18\(\)](#).

Here is the call graph for this function:



### 5.2.3.258 Rcpp\_lang20()

```

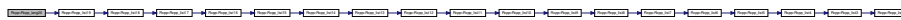
SEXP Rcpp::Rcpp_lang20 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8,
    SEXP x9,
    SEXP x10,
    SEXP x11,
    SEXP x12,
    SEXP x13,
    SEXP x14,
    SEXP x15,
    SEXP x16,
    SEXP x17,
    SEXP x18,
    SEXP x19 ) [inline]

```

Definition at line 287 of file lang.h.

References [Rcpp\\_list19\(\)](#).

Here is the call graph for this function:



### 5.2.3.259 Rcpp\_lang7()

```
SEXP Rcpp::Rcpp_lang7 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6 ) [inline]
```

Definition at line 79 of file lang.h.

References Rcpp\_list6().

Here is the call graph for this function:



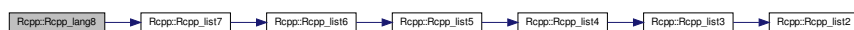
### 5.2.3.260 Rcpp\_lang8()

```
SEXP Rcpp::Rcpp_lang8 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7 ) [inline]
```

Definition at line 95 of file lang.h.

References Rcpp\_list7().

Here is the call graph for this function:



**5.2.3.261 Rcpp\_lang9()**

```
SEXP Rcpp::Rcpp_lang9 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8 ) [inline]
```

Definition at line 111 of file lang.h.

References [Rcpp\\_list8\(\)](#).

Here is the call graph for this function:

**5.2.3.262 Rcpp\_list10()**

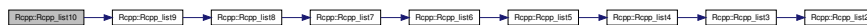
```
SEXP Rcpp::Rcpp_list10 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8,
    SEXP x9 ) [inline]
```

Definition at line 119 of file lang.h.

References [Rcpp\\_list9\(\)](#).

Referenced by [Rcpp\\_lang11\(\)](#), and [Rcpp\\_list11\(\)](#).

Here is the call graph for this function:



### 5.2.3.263 Rcpp\_list11()

```
SEXP Rcpp::Rcpp_list11 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8,
    SEXP x9,
    SEXP x10 ) [inline]
```

Definition at line 135 of file lang.h.

References [Rcpp\\_list10\(\)](#).

Referenced by [Rcpp\\_lang12\(\)](#), and [Rcpp\\_list12\(\)](#).

Here is the call graph for this function:



### 5.2.3.264 Rcpp\_list12()

```
SEXP Rcpp::Rcpp_list12 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8,
    SEXP x9,
    SEXP x10,
    SEXP x11 ) [inline]
```

Definition at line 151 of file lang.h.

References [Rcpp\\_list11\(\)](#).

Referenced by [Rcpp\\_lang13\(\)](#), and [Rcpp\\_list13\(\)](#).

Here is the call graph for this function:



### 5.2.3.265 Rcpp\_list13()

```
SEXP Rcpp::Rcpp_list13 (  
    SEXP x0,  
    SEXP x1,  
    SEXP x2,  
    SEXP x3,  
    SEXP x4,  
    SEXP x5,  
    SEXP x6,  
    SEXP x7,  
    SEXP x8,  
    SEXP x9,  
    SEXP x10,  
    SEXP x11,  
    SEXP x12 ) [inline]
```

Definition at line 167 of file lang.h.

References Rcpp\_list12().

Referenced by Rcpp\_lang14(), and Rcpp\_list14().

Here is the call graph for this function:



### 5.2.3.266 Rcpp\_list14()

```
SEXP Rcpp::Rcpp_list14 (  
    SEXP x0,  
    SEXP x1,  
    SEXP x2,  
    SEXP x3,  
    SEXP x4,  
    SEXP x5,  
    SEXP x6,  
    SEXP x7,  
    SEXP x8,  
    SEXP x9,  
    SEXP x10,  
    SEXP x11,  
    SEXP x12,  
    SEXP x13 ) [inline]
```

Definition at line 183 of file lang.h.

References Rcpp\_list13().

Referenced by Rcpp\_lang15(), and Rcpp\_list15().

Here is the call graph for this function:



### 5.2.3.267 Rcpp\_list15()

```

SEXP Rcpp::Rcpp_list15 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8,
    SEXP x9,
    SEXP x10,
    SEXP x11,
    SEXP x12,
    SEXP x13,
    SEXP x14 ) [inline]
  
```

Definition at line 199 of file lang.h.

References Rcpp\_list14().

Referenced by Rcpp\_lang16(), and Rcpp\_list16().

Here is the call graph for this function:





### 5.2.3.268 Rcpp\_list16()

```
SEXP Rcpp::Rcpp_list16 (  
    SEXP x0,  
    SEXP x1,  
    SEXP x2,  
    SEXP x3,  
    SEXP x4,  
    SEXP x5,  
    SEXP x6,  
    SEXP x7,  
    SEXP x8,  
    SEXP x9,  
    SEXP x10,  
    SEXP x11,  
    SEXP x12,  
    SEXP x13,  
    SEXP x14,  
    SEXP x15 ) [inline]
```

Definition at line 215 of file lang.h.

References [Rcpp\\_list15\(\)](#).

Referenced by [Rcpp\\_lang17\(\)](#), and [Rcpp\\_list17\(\)](#).

Here is the call graph for this function:



### 5.2.3.269 Rcpp\_list17()

```
SEXP Rcpp::Rcpp_list17 (  
    SEXP x0,  
    SEXP x1,  
    SEXP x2,  
    SEXP x3,  
    SEXP x4,  
    SEXP x5,  
    SEXP x6,  
    SEXP x7,  
    SEXP x8,  
    SEXP x9,  
    SEXP x10,  
    SEXP x11,  
    SEXP x12,  
    SEXP x13,
```

```

SEXP x14,
SEXP x15,
SEXP x16 ) [inline]

```

Definition at line 231 of file lang.h.

References `Rcpp_list16()`.

Referenced by `Rcpp_lang18()`, and `Rcpp_list18()`.

Here is the call graph for this function:



### 5.2.3.270 `Rcpp_list18()`

```

SEXP Rcpp::Rcpp_list18 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8,
    SEXP x9,
    SEXP x10,
    SEXP x11,
    SEXP x12,
    SEXP x13,
    SEXP x14,
    SEXP x15,
    SEXP x16,
    SEXP x17 ) [inline]

```

Definition at line 247 of file lang.h.

References `Rcpp_list17()`.

Referenced by `Rcpp_lang19()`, and `Rcpp_list19()`.

Here is the call graph for this function:



### 5.2.3.271 Rcpp\_list19()

```
SEXP Rcpp::Rcpp_list19 (  
    SEXP x0,  
    SEXP x1,  
    SEXP x2,  
    SEXP x3,  
    SEXP x4,  
    SEXP x5,  
    SEXP x6,  
    SEXP x7,  
    SEXP x8,  
    SEXP x9,  
    SEXP x10,  
    SEXP x11,  
    SEXP x12,  
    SEXP x13,  
    SEXP x14,  
    SEXP x15,  
    SEXP x16,  
    SEXP x17,  
    SEXP x18 ) [inline]
```

Definition at line 263 of file lang.h.

References [Rcpp\\_list18\(\)](#).

Referenced by [Rcpp\\_lang20\(\)](#), and [Rcpp\\_list20\(\)](#).

Here is the call graph for this function:



### 5.2.3.272 Rcpp\_list2()

```
SEXP Rcpp::Rcpp_list2 (  
    SEXP x0,  
    SEXP x1 ) [inline]
```

Definition at line 36 of file lang.h.

References [Rcpp\\_list1](#).

Referenced by [Rcpp\\_list3\(\)](#).

### 5.2.3.273 Rcpp\_list20()

```
SEXP Rcpp::Rcpp_list20 (  
    SEXP x0,  
    SEXP x1,  
    SEXP x2,  
    SEXP x3,  
    SEXP x4,  
    SEXP x5,  
    SEXP x6,  
    SEXP x7,  
    SEXP x8,  
    SEXP x9,  
    SEXP x10,  
    SEXP x11,  
    SEXP x12,  
    SEXP x13,  
    SEXP x14,  
    SEXP x15,  
    SEXP x16,  
    SEXP x17,  
    SEXP x18,  
    SEXP x19 ) [inline]
```

Definition at line 279 of file lang.h.

References Rcpp\_list19().

Here is the call graph for this function:



### 5.2.3.274 Rcpp\_list3()

```
SEXP Rcpp::Rcpp_list3 (  
    SEXP x0,  
    SEXP x1,  
    SEXP x2 ) [inline]
```

Definition at line 43 of file lang.h.

References Rcpp\_list2().

Referenced by Rcpp\_list4().

Here is the call graph for this function:



### 5.2.3.275 Rcpp\_list4()

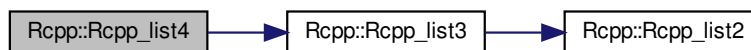
```
SEXP Rcpp::Rcpp_list4 (  
    SEXP x0,  
    SEXP x1,  
    SEXP x2,  
    SEXP x3 ) [inline]
```

Definition at line 50 of file lang.h.

References Rcpp\_list3().

Referenced by Rcpp\_list5().

Here is the call graph for this function:



### 5.2.3.276 Rcpp\_list5()

```
SEXP Rcpp::Rcpp_list5 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4 ) [inline]
```

Definition at line 57 of file lang.h.

References Rcpp\_list4().

Referenced by Rcpp\_list6().

Here is the call graph for this function:



### 5.2.3.277 Rcpp\_list6()

```
SEXP Rcpp::Rcpp_list6 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5 ) [inline]
```

Definition at line 64 of file lang.h.

References Rcpp\_list5().

Referenced by Rcpp\_lang7(), and Rcpp\_list7().

Here is the call graph for this function:



**5.2.3.278 Rcpp\_list7()**

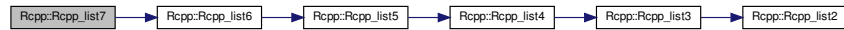
```
SEXP Rcpp::Rcpp_list7 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6 ) [inline]
```

Definition at line 71 of file lang.h.

References Rcpp\_list6().

Referenced by Rcpp\_lang8(), and Rcpp\_list8().

Here is the call graph for this function:

**5.2.3.279 Rcpp\_list8()**

```
SEXP Rcpp::Rcpp_list8 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7 ) [inline]
```

Definition at line 87 of file lang.h.

References Rcpp\_list7().

Referenced by Rcpp\_lang9(), and Rcpp\_list9().

Here is the call graph for this function:



### 5.2.3.280 Rcpp\_list9()

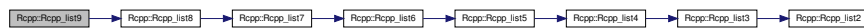
```
SEXP Rcpp::Rcpp_list9 (
    SEXP x0,
    SEXP x1,
    SEXP x2,
    SEXP x3,
    SEXP x4,
    SEXP x5,
    SEXP x6,
    SEXP x7,
    SEXP x8 ) [inline]
```

Definition at line 103 of file lang.h.

References [Rcpp\\_list8\(\)](#).

Referenced by [Rcpp\\_lang10\(\)](#), and [Rcpp\\_list10\(\)](#).

Here is the call graph for this function:



### 5.2.3.281 Rcpp\_namespace()

```
static Environment\_Impl Rcpp::Rcpp_namespace ( ) [static]
```

Returns

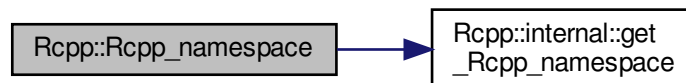
the [Rcpp](#) namespace

Definition at line 360 of file Environment.h.

References [Rcpp::internal::get\\_Rcpp\\_namespace\(\)](#).

Referenced by [Rcpp::internal::make\\_new\\_object\(\)](#), and [RCPP\\_API\\_CLASS\(\)](#).

Here is the call graph for this function:





**5.2.3.282 Rcpp\_precious\_init()**

```
void Rcpp::Rcpp_precious_init ( ) [inline]
```

Definition at line 142 of file routines.h.

References GET\_CALLABLE.

Referenced by R\_init\_Rcpp(), and registerFunctions().

**5.2.3.283 Rcpp\_precious\_preserve()**

```
SEXP Rcpp::Rcpp_precious_preserve (
    SEXP object ) [inline]
```

Definition at line 152 of file routines.h.

References GET\_CALLABLE.

Referenced by Rcpp\_PreciousPreserve(), and registerFunctions().

**5.2.3.284 Rcpp\_precious\_remove()**

```
void Rcpp::Rcpp_precious_remove (
    SEXP token ) [inline]
```

Definition at line 157 of file routines.h.

References GET\_CALLABLE.

Referenced by Rcpp\_PreciousRelease(), and registerFunctions().

**5.2.3.285 Rcpp\_precious\_teardown()**

```
void Rcpp::Rcpp_precious_teardown ( ) [inline]
```

Definition at line 147 of file routines.h.

References GET\_CALLABLE.

Referenced by R\_unload\_Rcpp(), and registerFunctions().

### 5.2.3.286 Rcpp\_PreciousPreserve()

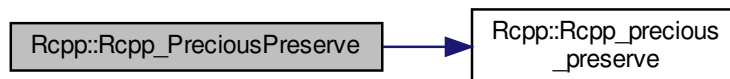
```
SEXP Rcpp::Rcpp_PreciousPreserve (
    SEXP object ) [inline]
```

Definition at line 116 of file RcppCommon.h.

References Rcpp\_precious\_preserve().

Referenced by Rcpp::String::append\_wide\_string(), Rcpp::String::assign\_wide\_string(), Rcpp::traits::named\_object<SEXP>::named\_object(), Rcpp::String::operator+=(), Rcpp::String::operator=(), Rcpp::PreserveStorage<CLASS>::set\_\_(), Rcpp::String::set\_encoding(), Rcpp::String::set\_na(), Rcpp::String::setData(), and Rcpp::String::String().

Here is the call graph for this function:



### 5.2.3.287 Rcpp\_PreciousRelease()

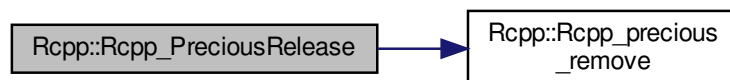
```
void Rcpp::Rcpp_PreciousRelease (
    SEXP token ) [inline]
```

Definition at line 120 of file RcppCommon.h.

References Rcpp\_precious\_remove().

Referenced by Rcpp::String::append\_wide\_string(), Rcpp::String::assign\_wide\_string(), Rcpp::PreserveStorage<CLASS>::invalidate\_\_(), Rcpp::String::operator+=(), Rcpp::String::operator=(), Rcpp::PreserveStorage<CLASS>::set\_\_(), Rcpp::String::set\_encoding(), Rcpp::String::set\_na(), Rcpp::traits::named\_object<SEXP>::~named\_object(), Rcpp::PreserveStorage<CLASS>::~PreserveStorage(), and Rcpp::String::~String().

Here is the call graph for this function:



### 5.2.3.288 Rcpp\_PreserveObject()

```
SEXP Rcpp::Rcpp_PreserveObject (
    SEXP x ) [inline]
```

Definition at line 98 of file RcppCommon.h.

Referenced by Rcpp\_ReplaceObject().

### 5.2.3.289 Rcpp\_protect()

```
SEXP Rcpp::Rcpp_protect (
    SEXP x ) [inline]
```

Definition at line 23 of file Shield.h.

Referenced by Rcpp::Shelter< T >::operator()().

### 5.2.3.290 Rcpp\_ReleaseObject()

```
void Rcpp::Rcpp_ReleaseObject (
    SEXP x ) [inline]
```

Definition at line 102 of file RcppCommon.h.

Referenced by Rcpp\_ReplaceObject().

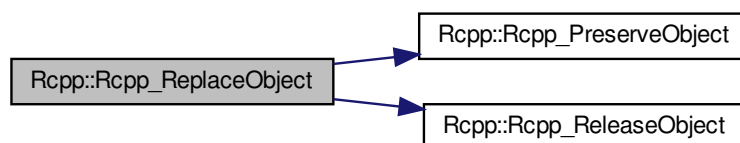
### 5.2.3.291 Rcpp\_ReplaceObject()

```
SEXP Rcpp::Rcpp_ReplaceObject (
    SEXP x,
    SEXP y ) [inline]
```

Definition at line 105 of file RcppCommon.h.

References Rcpp\_PreserveObject(), and Rcpp\_ReleaseObject().

Here is the call graph for this function:



### 5.2.3.292 RCPP\_SIMPLE\_EXCEPTION\_CLASS()

```
Rcpp::RCPP_SIMPLE_EXCEPTION_CLASS (
    not_a_matrix ,
    "Not a matrix." )
```

### 5.2.3.293 Rcpp\_unprotect()

```
void Rcpp::Rcpp_unprotect (
    int i ) [inline]
```

Definition at line 28 of file Shield.h.

Referenced by Rcpp::Shelter< T >::~~Shelter(), and Rcpp::Shield< T >::~~Shield().

### 5.2.3.294 remove()

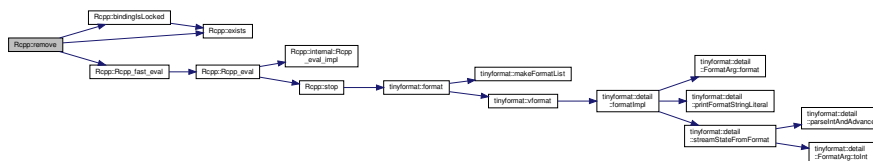
```
bool Rcpp::remove (
    const std::string & name )
```

remove an object from this environment

Definition at line 240 of file Environment.h.

References bindingsLocked(), exists(), and Rcpp\_fast\_eval().

Here is the call graph for this function:



### 5.2.3.295 rep() [1/6]

```
sugar::Rep_Single<bool> Rcpp::rep (
    const bool & x,
    R_xlen_t n ) [inline]
```

Definition at line 85 of file rep.h.

**5.2.3.296 rep()** [2/6]

```
sugar::Rep_Single<double> Rcpp::rep (
    const double & x,
    R_xlen_t n ) [inline]
```

Definition at line 73 of file rep.h.

**5.2.3.297 rep()** [3/6]

```
sugar::Rep_Single<int> Rcpp::rep (
    const int & x,
    R_xlen_t n ) [inline]
```

Definition at line 76 of file rep.h.

**5.2.3.298 rep()** [4/6]

```
sugar::Rep_Single<Rbyte> Rcpp::rep (
    const Rbyte & x,
    R_xlen_t n ) [inline]
```

Definition at line 79 of file rep.h.

**5.2.3.299 rep()** [5/6]

```
sugar::Rep_Single<Rcomplex> Rcpp::rep (
    const Rcomplex & x,
    R_xlen_t n ) [inline]
```

Definition at line 82 of file rep.h.

**5.2.3.300 rep()** [6/6]

```
template<int RTYPE, bool NA, typename T >
sugar::Rep<RTYPE,NA,T> Rcpp::rep (
    const VectorBase< RTYPE, NA, T > & t,
    R_xlen_t n ) [inline]
```

Definition at line 69 of file rep.h.

### 5.2.3.301 rep\_each()

```
template<int RTYPE, bool NA, typename T >
sugar::Rep_each<RTYPE,NA,T> Rcpp::rep_each (
    const VectorBase< RTYPE, NA, T > & t,
    R_xlen_t times ) [inline]
```

Definition at line 50 of file rep\_each.h.

### 5.2.3.302 rep\_len()

```
template<int RTYPE, bool NA, typename T >
sugar::Rep_len<RTYPE,NA,T> Rcpp::rep_len (
    const VectorBase< RTYPE, NA, T > & t,
    R_xlen_t len ) [inline]
```

Definition at line 50 of file rep\_len.h.

### 5.2.3.303 rev()

```
template<int RTYPE, bool NA, typename T >
sugar::Rev<RTYPE,NA,T> Rcpp::rev (
    const VectorBase< RTYPE, NA, T > & t ) [inline]
```

Definition at line 50 of file rev.h.

### 5.2.3.304 rexp() [1/2]

```
NumericVector Rcpp::rexp (
    int n ) [inline]
```

Definition at line 147 of file random.h.

### 5.2.3.305 rexp() [2/2]

```
NumericVector Rcpp::rexp (
    int n,
    double rate ) [inline]
```

Definition at line 137 of file random.h.

### 5.2.3.306 rf()

```
NumericVector Rcpp::rf (  
    int n,  
    double n1,  
    double n2 ) [inline]
```

Definition at line 151 of file random.h.

### 5.2.3.307 rgamma() [1/2]

```
NumericVector Rcpp::rgamma (  
    int n,  
    double a ) [inline]
```

Definition at line 174 of file random.h.

### 5.2.3.308 rgamma() [2/2]

```
NumericVector Rcpp::rgamma (  
    int n,  
    double a,  
    double scale ) [inline]
```

Definition at line 165 of file random.h.

### 5.2.3.309 rgeom()

```
NumericVector Rcpp::rgeom (  
    int n,  
    double p ) [inline]
```

Definition at line 183 of file random.h.

### 5.2.3.310 rhyper()

```
NumericVector Rcpp::rhyper (
    int n,
    double nn1,
    double nn2,
    double kk ) [inline]
```

Definition at line 189 of file random.h.

### 5.2.3.311 rlnorm() [1/3]

```
NumericVector Rcpp::rlnorm (
    int n ) [inline]
```

Definition at line 215 of file random.h.

### 5.2.3.312 rlnorm() [2/3]

```
NumericVector Rcpp::rlnorm (
    int n,
    double meanlog ) [inline]
```

Definition at line 204 of file random.h.

References `Rcpp::algorithm::exp()`.

Here is the call graph for this function:





**5.2.3.313 rlnorm()** [3/3]

```
NumericVector Rcpp::rlnorm (  
    int n,  
    double meanlog,  
    double sdlog ) [inline]
```

Definition at line 193 of file random.h.

References Rcpp::algorithm::exp().

Here is the call graph for this function:

**5.2.3.314 rlogis()** [1/3]

```
NumericVector Rcpp::rlogis (  
    int n ) [inline]
```

Definition at line 239 of file random.h.

**5.2.3.315 rlogis()** [2/3]

```
NumericVector Rcpp::rlogis (  
    int n,  
    double location ) [inline]
```

Definition at line 229 of file random.h.

### 5.2.3.316 `rlogis()` [3/3]

```
NumericVector Rcpp::rlogis (  
    int n,  
    double location,  
    double scale ) [inline]
```

Definition at line 219 of file `random.h`.

### 5.2.3.317 `rnbinom()`

```
NumericVector Rcpp::rnbinom (  
    int n,  
    double siz,  
    double prob ) [inline]
```

Definition at line 243 of file `random.h`.

### 5.2.3.318 `rnbinom_mu()`

```
NumericVector Rcpp::rnbinom_mu (  
    int n,  
    double siz,  
    double mu ) [inline]
```

Definition at line 251 of file `random.h`.

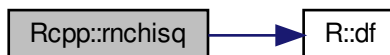
### 5.2.3.319 `rnchisq()` [1/2]

```
NumericVector Rcpp::rnchisq (  
    int n,  
    double df ) [inline]
```

Definition at line 268 of file `random.h`.

References `R::df()`.

Here is the call graph for this function:



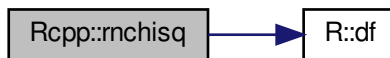
**5.2.3.320 rchisq()** [2/2]

```
NumericVector Rcpp::rchisq (  
    int n,  
    double df,  
    double lambda ) [inline]
```

Definition at line 258 of file random.h.

References R::df().

Here is the call graph for this function:

**5.2.3.321 rnorm()** [1/3]

```
NumericVector Rcpp::rnorm (  
    int n ) [inline]
```

Definition at line 96 of file random.h.

**5.2.3.322 rnorm()** [2/3]

```
NumericVector Rcpp::rnorm (  
    int n,  
    double mean ) [inline]
```

Definition at line 80 of file random.h.

References mean().

Here is the call graph for this function:



**5.2.3.323 rnorm()** [3/3]

```

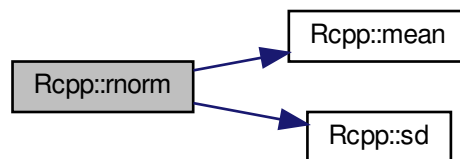
NumericVector Rcpp::rnorm (
    int n,
    double mean,
    double sd ) [inline]

```

Definition at line 58 of file random.h.

References `mean()`, and `sd()`.

Here is the call graph for this function:

**5.2.3.324 row()**

```

template<int RTYPE, bool LHS_NA, typename LHS_T >
sugar::Row<RTYPE, LHS_NA, LHS_T> Rcpp::row (
    const Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T > & lhs ) [inline]

```

Definition at line 55 of file row.h.

Referenced by operator<<().

**5.2.3.325 rowMeans()**

```

template<int RTYPE, bool NA, typename T >
sugar::detail::RowMeansReturn<RTYPE>::type Rcpp::rowMeans (
    const MatrixBase< RTYPE, NA, T > & x,
    bool na_rm = false ) [inline]

```

Definition at line 942 of file rowSums.h.

### 5.2.3.326 rownames()

```
internal::DimNameProxy Rcpp::rownames (  
    SEXP x ) [inline]
```

Definition at line 209 of file Matrix.h.

### 5.2.3.327 rowSums()

```
template<int RTYPE, bool NA, typename T >  
sugar::detail::RowSumsReturn<RTYPE>::type Rcpp::rowSums (  
    const MatrixBase< RTYPE, NA, T > & x,  
    bool na_rm = false ) [inline]
```

Definition at line 924 of file rowSums.h.

### 5.2.3.328 rpois()

```
NumericVector Rcpp::rpois (  
    int n,  
    double mu ) [inline]
```

Definition at line 274 of file random.h.

### 5.2.3.329 rsignrank()

```
NumericVector Rcpp::rsignrank (  
    int n,  
    double nn ) [inline]
```

Definition at line 278 of file random.h.

### 5.2.3.330 `rt()`

```
NumericVector Rcpp::rt (  
    int n,  
    double df ) [inline]
```

Definition at line 282 of file `random.h`.

References `R::df()`.

Here is the call graph for this function:



### 5.2.3.331 `runif()` [1/3]

```
NumericVector Rcpp::runif (  
    int n ) [inline]
```

Definition at line 307 of file `random.h`.

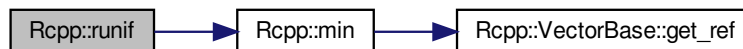
### 5.2.3.332 `runif()` [2/3]

```
NumericVector Rcpp::runif (  
    int n,  
    double min ) [inline]
```

Definition at line 301 of file `random.h`.

References `min()`.

Here is the call graph for this function:



**5.2.3.333** `runif()` [3/3]

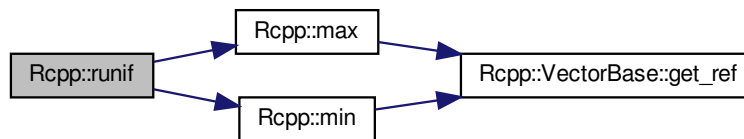
```
NumericVector Rcpp::runif (
    int n,
    double min,
    double max ) [inline]
```

Definition at line 295 of file random.h.

References `max()`, and `min()`.

Referenced by `piSugar()`.

Here is the call graph for this function:

**5.2.3.334** `rweibull()` [1/2]

```
NumericVector Rcpp::rweibull (
    int n,
    double shape ) [inline]
```

Definition at line 320 of file random.h.

**5.2.3.335** `rweibull()` [2/2]

```
NumericVector Rcpp::rweibull (
    int n,
    double shape,
    double scale ) [inline]
```

Definition at line 311 of file random.h.

### 5.2.3.336 `rwilcox()`

```

NumericVector Rcpp::rwilcox (
    int n,
    double mm,
    double nn ) [inline]

```

Definition at line 327 of file random.h.

### 5.2.3.337 `sample()` [1/2]

```

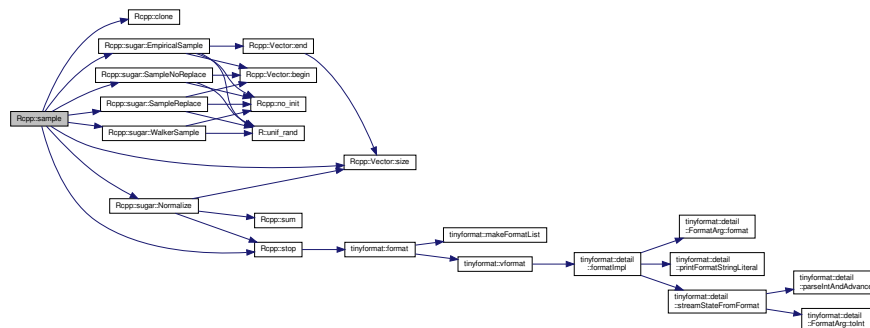
template<int RTYPE>
Vector<RTYPE> Rcpp::sample (
    const Vector< RTYPE > & x,
    int size,
    bool replace = false,
    sugar::probs_t probs = R_NilValue ) [inline]

```

Definition at line 437 of file sample.h.

References `clone()`, `Rcpp::sugar::EmpiricalSample()`, `Rcpp::sugar::Normalize()`, `Rcpp::sugar::SampleNoReplace()`, `Rcpp::sugar::SampleReplace()`, `Rcpp::Vector< RTYPE, StoragePolicy >::size()`, `stop()`, and `Rcpp::sugar::Walker`↔`Sample()`.

Here is the call graph for this function:





5.2.3.338 `sample()` [2/2]

```

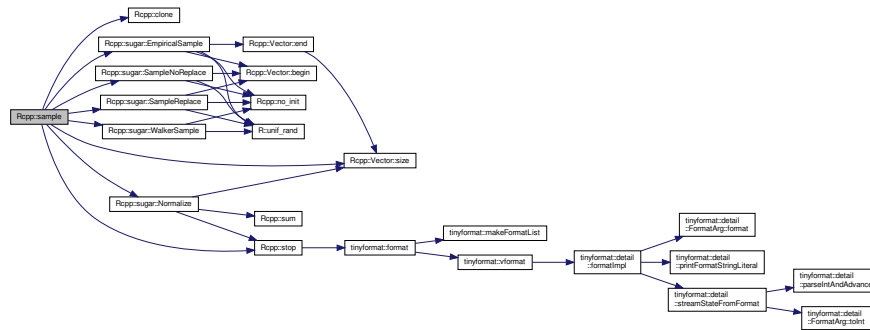
Vector<INTSXP> Rcpp::sample (
    int n,
    int size,
    bool replace = false,
    sugar::probs_t probs = R_NilValue,
    bool one_based = true ) [inline]

```

Definition at line 401 of file `sample.h`.

References `clone()`, `Rcpp::sugar::EmpiricalSample()`, `Rcpp::sugar::Normalize()`, `Rcpp::sugar::SampleNoReplace()`, `Rcpp::sugar::SampleReplace()`, `Rcpp::Vector< RTYPE, StoragePolicy >::size()`, `stop()`, and `Rcpp::sugar::Walker←Sample()`.

Here is the call graph for this function:

5.2.3.339 `sapply()` [1/2]

```

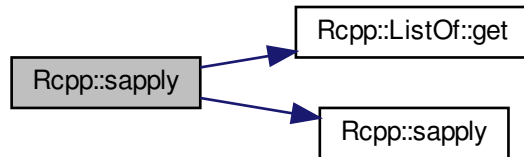
template<typename T , typename Function >
T Rcpp::sapply (
    const ListOf< T > & t,
    Function fun )

```

Definition at line 134 of file `ListOf.h`.

References `Rcpp::ListOf< T >::get()`, and `sapply()`.

Here is the call graph for this function:



### 5.2.3.340 sapply() [2/2]

```

template<int RTYPE, bool NA, typename T , typename Function >
sugar::Sapply< RTYPE,NA,T,Function, traits::same_type< typename ::Rcpp::sugar::sapply_application_result_of<Func
T>::type , typename Rcpp::traits::storage_type< traits::r_sexptype_traits< typename ::Rcpp::sugar::sapply_applic
T>::type >::rtype >::type >::value> Rcpp::sapply (
    const Rcpp::VectorBase< RTYPE, NA, T > & t,
    Function fun ) [inline]
  
```

Definition at line 126 of file sapply.h.

References NA.

Referenced by sapply().

### 5.2.3.341 sd()

```

template<bool NA, typename T >
sugar::Sd<REALSXP,NA,T> Rcpp::sd (
    const VectorBase< REALSXP, NA, T > & t ) [inline]
  
```

Definition at line 46 of file sd.h.

Referenced by rnorm().

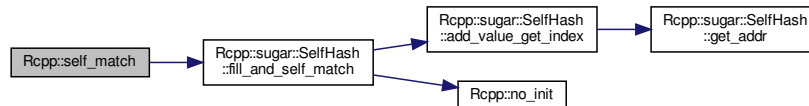
### 5.2.3.342 self\_match()

```
template<int RTYPE, bool NA, typename T >
IntegerVector Rcpp::self_match (
    const VectorBase< RTYPE, NA, T > & x ) [inline]
```

Definition at line 69 of file self\_match.h.

References Rcpp::sugar::SelfHash< RTYPE >::fill\_and\_self\_match().

Here is the call graph for this function:



### 5.2.3.343 seq()

```
Range Rcpp::seq (
    R_xlen_t start,
    R_xlen_t end ) [inline]
```

Definition at line 52 of file seq\_along.h.

### 5.2.3.344 seq\_along()

```
template<int RTYPE, bool NA, typename T >
sugar::SeqLen Rcpp::seq_along (
    const Rcpp::VectorBase< RTYPE, NA, T > & t ) [inline]
```

Definition at line 44 of file seq\_along.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



### 5.2.3.345 seq\_len()

```
sugar::SeqLen Rcpp::seq_len (
    const size_t & n ) [inline]
```

Definition at line 48 of file seq\_along.h.

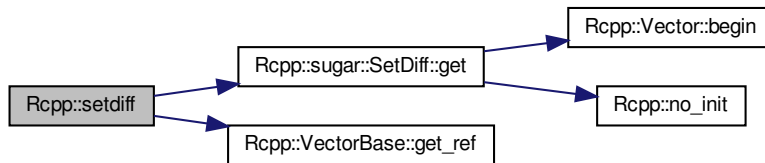
### 5.2.3.346 setdiff()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Vector<RTYPE> Rcpp::setdiff (
    const VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 161 of file setdiff.h.

References `Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get()`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref()`.

Here is the call graph for this function:



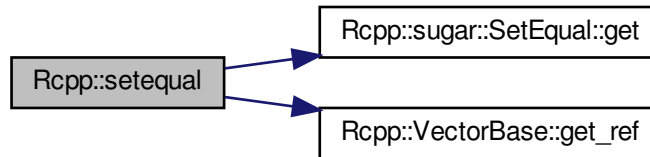
### 5.2.3.347 setequal()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
bool Rcpp::setequal (
    const VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 166 of file setdiff.h.

References `Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get()`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref()`.

Here is the call graph for this function:



### 5.2.3.348 setFunction()

```
void Rcpp::setFunction (
    const Function & function )
```

sets the function

Definition at line 126 of file Language.h.

### 5.2.3.349 setSymbol() [1/2]

```
void Rcpp::setSymbol (
    const std::string & symbol )
```

Creates a call to the given symbol using variable number of arguments

#### Parameters

<i>symbol</i>	symbol
<i>...Args</i>	variable length argument list. The type of each argument must be wrappable, meaning there need to be a wrap function that takes this type as its parameter

For example, `Language( "rnorm", 10, 0.0 )` will create the same call as

```
call( "rnorm", 10L, 0.0 )
```

10 is wrapped as an integer vector using `wrap( const& int )` 0.0 is wrapped as a numeric vector using `wrap( const& double )` ... sets the symbol of the call

Definition at line 110 of file Language.h.

### 5.2.3.350 setSymbol() [2/2]

```
void Rcpp::setSymbol (
    const Symbol & symbol )
```

sets the symbol of the call

Definition at line 117 of file Language.h.

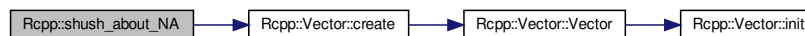
### 5.2.3.351 shush\_about\_NA()

```
LogicalVector Rcpp::shush_about_NA ( ) [inline]
```

Definition at line 54 of file Na\_Proxy.h.

References `_`, `Rcpp::Vector< RTYPE, StoragePolicy >::create()`, and `NA`.

Here is the call graph for this function:



### 5.2.3.352 sign() [1/2]

```
template<bool NA, typename T >
sugar::Sign<INTSXP,NA,T> Rcpp::sign (
    const VectorBase< INTSXP, NA, T > & t ) [inline]
```

Definition at line 70 of file sign.h.

Referenced by `Rcpp::Module::get_function()`.

**5.2.3.353 sign()** [2/2]

```
template<bool NA, typename T >
sugar::Sign<REALSXP,NA,T> Rcpp::sign (
    const VectorBase< REALSXP, NA, T > & t ) [inline]
```

Definition at line 75 of file sign.h.

**5.2.3.354 sort\_unique()**

```
template<int RTYPE, bool NA, typename T >
Vector<RTYPE> Rcpp::sort_unique (
    const VectorBase< RTYPE, NA, T > & t,
    bool decreasing = false ) [inline]
```

Definition at line 72 of file unique.h.

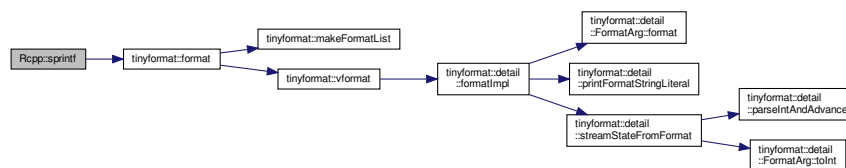
**5.2.3.355 sprintf()**

```
template<int MAX_SIZE>
std::string Rcpp::sprintf (
    const char * format,
    ... )
```

Definition at line 28 of file sprintf.h.

References `tinyformat::format()`.

Here is the call graph for this function:



### 5.2.3.356 standard\_delete\_finalizer()

```
template<typename T >
void Rcpp::standard_delete_finalizer (
    T * obj )
```

Definition at line 30 of file XPtr.h.

### 5.2.3.357 stop() [1/12]

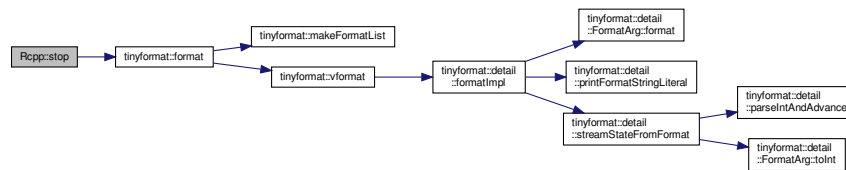
```
template<typename... Args>
void NORET Rcpp::stop (
    const char * fmt,
    Args &&... args ) [inline]
```

Definition at line 51 of file exceptions.h.

References tinyformat::format().

Referenced by Rcpp::internal::DimNameProxy::assign(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::check\_indices(), Rcpp::Vector< RTYPE, StoragePolicy >::findName(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::get\_indices(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::JoinOp(), Rcpp::sugar::Normalize(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::operator=(), RCPP\_API\_CLASS(), Rcpp\_eval(), sample(), and trimws().

Here is the call graph for this function:





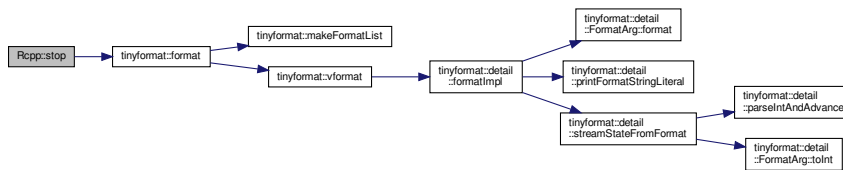
## 5.2.3.358 stop() [2/12]

```
template<typename T1 >
void NORET Rcpp::stop (
    const char * fmt,
    const T1 & arg1 ) [inline]
```

Definition at line 126 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



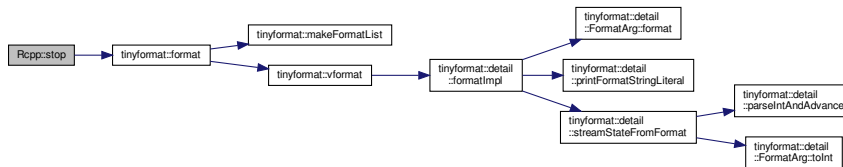
## 5.2.3.359 stop() [3/12]

```
template<typename T1 , typename T2 >
void NORET Rcpp::stop (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2 ) [inline]
```

Definition at line 131 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



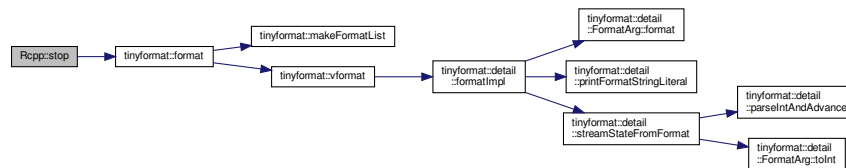
**5.2.3.360 stop()** [4/12]

```
template<typename T1 , typename T2 , typename T3 >
void NORET Rcpp::stop (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3 ) [inline]
```

Definition at line 136 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:

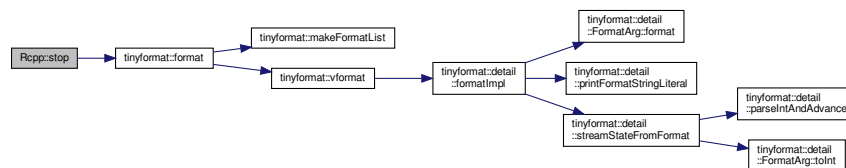
**5.2.3.361 stop()** [5/12]

```
template<typename T1 , typename T2 , typename T3 , typename T4 >
void NORET Rcpp::stop (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4 ) [inline]
```

Definition at line 141 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



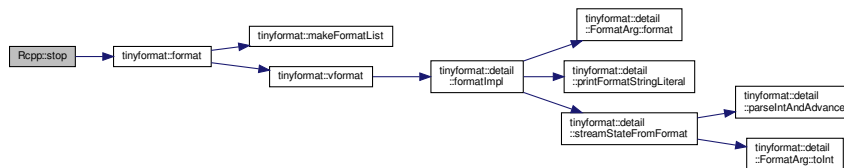
## 5.2.3.362 stop() [6/12]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 >
void NORET Rcpp::stop (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4,
    const T5 & arg5 ) [inline]
```

Definition at line 146 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



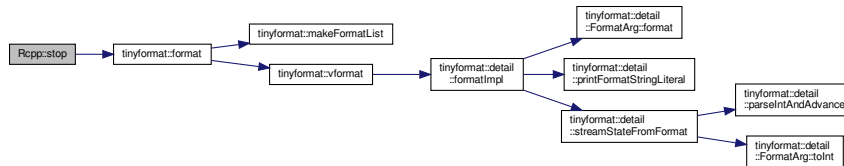
## 5.2.3.363 stop() [7/12]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 >
void NORET Rcpp::stop (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4,
    const T5 & arg5,
    const T6 & arg6 ) [inline]
```

Definition at line 151 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



**5.2.3.364 stop()** [8/12]

```

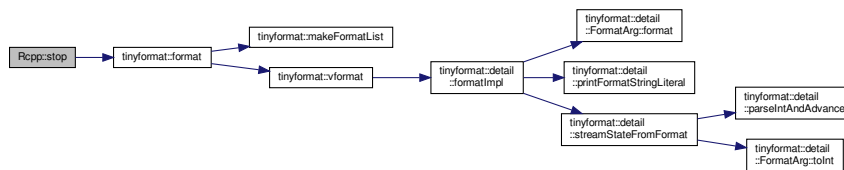
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 >
void NORET Rcpp::stop (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4,
    const T5 & arg5,
    const T6 & arg6,
    const T7 & arg7 ) [inline]

```

Definition at line 156 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:

**5.2.3.365 stop()** [9/12]

```

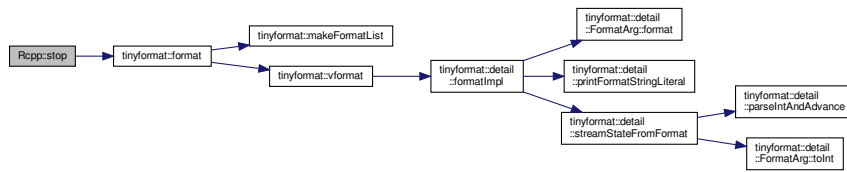
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 >
void NORET Rcpp::stop (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4,
    const T5 & arg5,
    const T6 & arg6,
    const T7 & arg7,
    const T8 & arg8 ) [inline]

```

Definition at line 161 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



### 5.2.3.366 stop() [10/12]

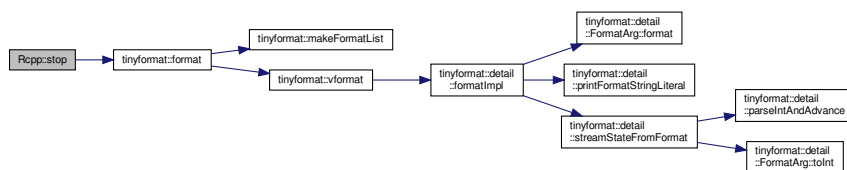
```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 >
```

```
void NORET Rcpp::stop (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4,
    const T5 & arg5,
    const T6 & arg6,
    const T7 & arg7,
    const T8 & arg8,
    const T9 & arg9 ) [inline]
```

Definition at line 166 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



**5.2.3.367 stop()** [11/12]

```

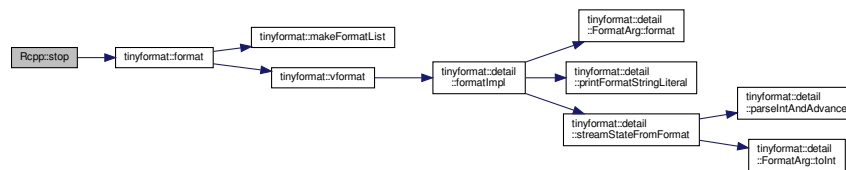
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 >
void NORET Rcpp::stop (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4,
    const T5 & arg5,
    const T6 & arg6,
    const T7 & arg7,
    const T8 & arg8,
    const T9 & arg9,
    const T10 & arg10 ) [inline]

```

Definition at line 171 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:

**5.2.3.368 stop()** [12/12]

```

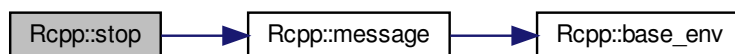
void Rcpp::stop (
    const std::string & message ) [inline]

```

Definition at line 117 of file exceptions.h.

References `message()`.

Here is the call graph for this function:



**5.2.3.369 StretchyList\_Impl()** [1/2]

```
Rcpp::StretchyList_Impl ( )
```

Definition at line 42 of file StretchyList.h.

**5.2.3.370 StretchyList\_Impl()** [2/2]

```
Rcpp::StretchyList_Impl (
    SEXP x )
```

Definition at line 47 of file StretchyList.h.

**5.2.3.371 sum()** [1/3]

```
template<bool NA, typename T >
sugar::Sum<INTSXP,NA,T> Rcpp::sum (
    const VectorBase< INTSXP, NA, T > & t ) [inline]
```

Definition at line 98 of file sum.h.

Referenced by Rcpp::sugar::na\_omit\_impl(), Rcpp::sugar::Normalize(), and piSugar().

**5.2.3.372 sum()** [2/3]

```
template<bool NA, typename T >
sugar::Sum<LGLSXP,NA,T> Rcpp::sum (
    const VectorBase< LGLSXP, NA, T > & t ) [inline]
```

Definition at line 108 of file sum.h.

**5.2.3.373 sum()** [3/3]

```
template<bool NA, typename T >
sugar::Sum<REALSXP,NA,T> Rcpp::sum (
    const VectorBase< REALSXP, NA, T > & t ) [inline]
```

Definition at line 103 of file sum.h.

**5.2.3.374 table()**

```
template<int RTYPE, bool NA, typename T >
IntegerVector Rcpp::table (
    const VectorBase< RTYPE, NA, T > & x ) [inline]
```

Definition at line 126 of file table.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref().

Referenced by in(), match(), Rcpp::sugar::SelfMatch< RTYPE, TABLE\_T >::SelfMatch(), and Rcpp::sugar::Table< RTYPE, TABLE\_T >::Table().

Here is the call graph for this function:

**5.2.3.375 tail()**

```
template<int RTYPE, bool NA, typename T >
sugar::Tail<RTYPE,NA,T> Rcpp::tail (
    const VectorBase< RTYPE, NA, T > & t,
    R_xlen_t n ) [inline]
```

Definition at line 56 of file tail.h.

Referenced by grow(), Rcpp::internal::grow\_\_dispatch(), Rcpp::DottedPairImpl< CLASS >::insert(), and Rcpp::↔DottedPairImpl< CLASS >::push\_back().



**5.2.3.376 timesub()**

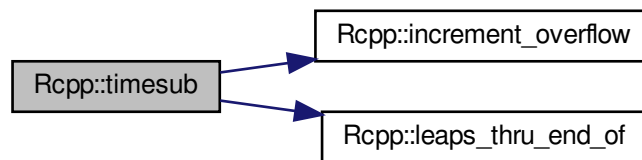
```
static struct tm * Rcpp::timesub (
    const time_t * timep,
    int_fast32_t offset,
    const struct state * sp,
    struct tm * tmp ) [static]
```

Definition at line 1227 of file date.cpp.

References `DAYSPERLYEAR`, `DAYSPEERYEAR`, `DAYSPEERWEEK`, `EPOCH_WDAY`, `EPOCH_YEAR`, `increment_overflow()`, `isleap`, `Rcpp::state::leapcnt`, `leaps_thru_end_of()`, `Rcpp::lsinfo::ls_corr`, `Rcpp::lsinfo::ls_trans`, `Rcpp::state::::lsis`, `mon_lengths`, `SECSPERDAY`, `SECSPEERHOUR`, `SECSPEERMIN`, `TM_YEAR_BASE`, `TYPE_SIGNED`, and `year_lengths`.

Referenced by `gmtsub()`.

Here is the call graph for this function:

**5.2.3.377 toString()**

```
static std::string Rcpp::toString (
    const int i ) [static]
```

Definition at line 64 of file exceptions.h.

**5.2.3.378 tranpose\_impl()**

```
template<int RTYPE, template< class > class StoragePolicy>
Matrix<RTYPE, StoragePolicy> Rcpp::tranpose_impl (
    const Matrix< RTYPE, StoragePolicy > & x )
```

Definition at line 406 of file Matrix.h.

### 5.2.3.379 transpose() [1/3]

```
template<template< class > class StoragePolicy>
Matrix<INTSXP, StoragePolicy> Rcpp::transpose (
    const Matrix< INTSXP, StoragePolicy > & x )
```

Definition at line 440 of file Matrix.h.

### 5.2.3.380 transpose() [2/3]

```
template<template< class > class StoragePolicy>
Matrix<REALSXP, StoragePolicy> Rcpp::transpose (
    const Matrix< REALSXP, StoragePolicy > & x )
```

Definition at line 435 of file Matrix.h.

### 5.2.3.381 transpose() [3/3]

```
template<template< class > class StoragePolicy>
Matrix<STRSXP, StoragePolicy> Rcpp::transpose (
    const Matrix< STRSXP, StoragePolicy > & x )
```

Definition at line 445 of file Matrix.h.

### 5.2.3.382 transtime()

```
static int_fast32_t Rcpp::transtime (
    int year,
    const struct rule * rulep,
    int_fast32_t offset ) [static]
```

Definition at line 879 of file date.cpp.

References DAY\_OF\_YEAR, DAYSPERWEEK, INITIALIZE, isleap, JULIAN\_DAY, mon\_lengths, MONTH\_NTH\_DAY\_OF\_WEEK, Rcpp::rule::r\_day, Rcpp::rule::r\_mon, Rcpp::rule::r\_time, Rcpp::rule::r\_type, Rcpp::rule::r\_week, and SECSPERDAY.

Referenced by tzparse().

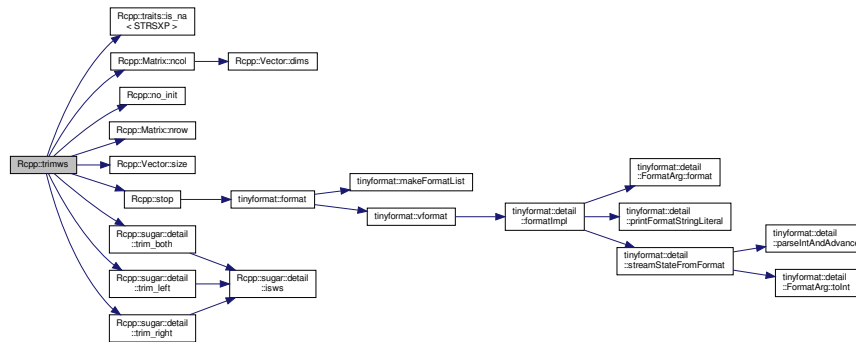
## 5.2.3.383 trimws() [1/3]

```
Matrix<STRSXP> Rcpp::trimws (
    const Matrix< STRSXP > & x,
    const char * which = "both" ) [inline]
```

Definition at line 135 of file trimws.h.

References Rcpp::traits::is\_na< STRSXP >(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), no\_init(), Rcpp::Matrix< RTYPE, StoragePolicy >::nrow(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), stop(), Rcpp::sugar::detail::trim\_both(), Rcpp::sugar::detail::trim\_left(), and Rcpp::sugar::detail::trim\_right().

Here is the call graph for this function:



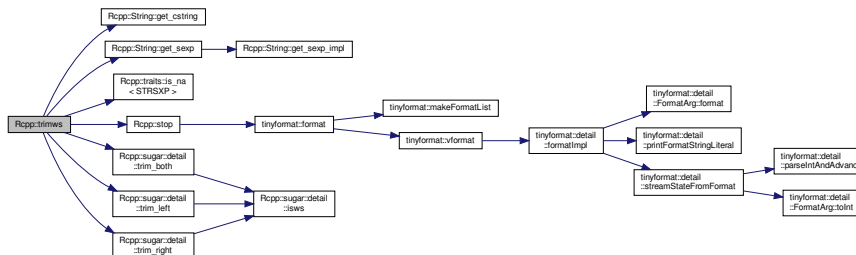
## 5.2.3.384 trimws() [2/3]

```
String Rcpp::trimws (
    const String & str,
    const char * which = "both" ) [inline]
```

Definition at line 181 of file trimws.h.

References Rcpp::String::get\_cstring(), Rcpp::String::get\_sexp(), Rcpp::traits::is\_na< STRSXP >(), stop(), Rcpp::sugar::detail::trim\_both(), Rcpp::sugar::detail::trim\_left(), and Rcpp::sugar::detail::trim\_right().

Here is the call graph for this function:



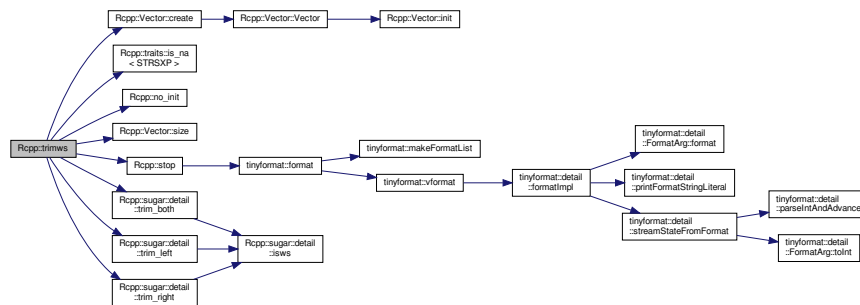
### 5.2.3.385 trimws() [3/3]

```
Vector<STRSXP> Rcpp::trimws (
    const Vector< STRSXP > & x,
    const char * which = "both" ) [inline]
```

Definition at line 90 of file trimws.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::create(), Rcpp::traits::is\_na< STRSXP >(), no\_init(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), stop(), Rcpp::sugar::detail::trim\_both(), Rcpp::sugar::detail::trim\_left(), and Rcpp::sugar::detail::trim\_right().

Here is the call graph for this function:



### 5.2.3.386 type2name()

```
const char * Rcpp::type2name (
    SEXP x ) [inline]
```

Definition at line 84 of file routines.h.

References GET\_CALLABLE.

Referenced by registerFunctions().

### 5.2.3.387 typesequiv()

```
static int Rcpp::typesequiv (
    const struct state * sp,
    int a,
    int b ) [static]
```

Definition at line 1202 of file date.cpp.

References Rcpp::state::chars, Rcpp::tinfo::tt\_abbrind, Rcpp::tinfo::tt\_gmtoff, Rcpp::tinfo::tt\_isdst, Rcpp::tinfo::tt\_←ttisgmt, Rcpp::tinfo::tt\_tisstd, Rcpp::state::ttis, and Rcpp::state::typecnt.

Referenced by tzload().

## 5.2.3.388 tzload()

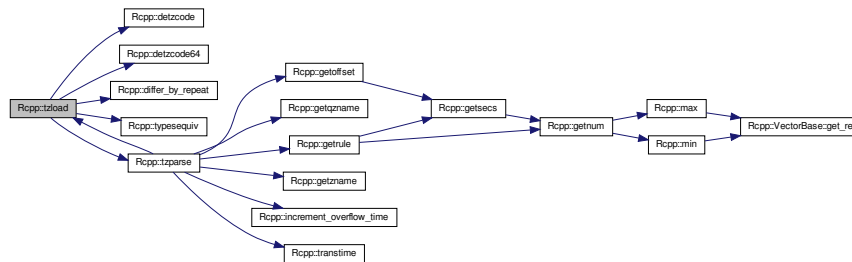
```
static int Rcpp::tzload (
    const char * name,
    struct state *const sp,
    const int doextend ) [static]
```

Definition at line 632 of file date.cpp.

References Rcpp::state::ats, Rcpp::state::charcnt, Rcpp::state::chars, detzcode(), detzcode64(), differ\_by\_repeat(), Rcpp::state::goahead, Rcpp::state::goback, Rcpp::state::leapcnt, Rcpp::linfo::ls\_corr, Rcpp::linfo::ls\_trans, Rcpp::state::lisis, OPEN\_MODE, Rcpp::state::timecnt, Rcpp::tinfo::tt\_abbrind, Rcpp::tinfo::tt\_gmtoff, Rcpp::tinfo::tt\_isdst, Rcpp::tinfo::tt\_tisgmt, Rcpp::tinfo::tt\_tisstd, Rcpp::state::ttis, TYPE\_INTEGRAL, TYPE\_SIGNED, Rcpp::state::typecnt, Rcpp::state::types, typesequiv(), TZ\_MAX\_CHARS, TZ\_MAX\_LEAPS, TZ\_MAX\_TIMES, TZ\_MAX\_TYPES, TZDEFAULT, tzparse(), and YEARSPPERREPEAT.

Referenced by gmtload(), and tzparse().

Here is the call graph for this function:



## 5.2.3.389 tzparse()

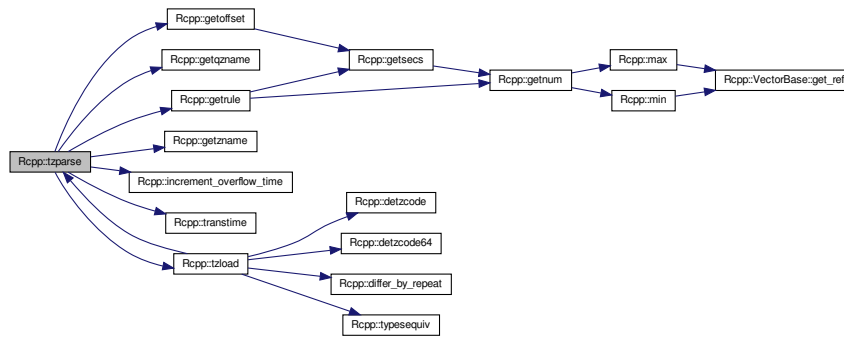
```
static int Rcpp::tzparse (
    const char * name,
    struct state * sp,
    int lastditch ) [static]
```

Definition at line 966 of file date.cpp.

References Rcpp::state::ats, Rcpp::state::charcnt, Rcpp::state::chars, EPOCH\_YEAR, getoffset(), getzname(), getrule(), getzname(), increment\_overflow\_time(), INITIALIZE, isleap, Rcpp::state::leapcnt, SECSPERDAY, SECSPERHOUR, Rcpp::state::timecnt, transtime(), Rcpp::tinfo::tt\_abbrind, Rcpp::tinfo::tt\_gmtoff, Rcpp::tinfo::tt\_isdst, Rcpp::tinfo::tt\_tisgmt, Rcpp::tinfo::tt\_tisstd, Rcpp::state::ttis, Rcpp::state::typecnt, Rcpp::state::types, TZ\_MAX\_TIMES, TZDEFRULES, TZDEFRULESTRING, tzload(), year\_lengths, and YEARSPPERREPEAT.

Referenced by gmtload(), and tzload().

Here is the call graph for this function:



### 5.2.3.390 union\_()

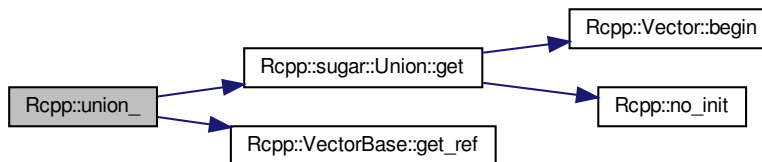
```

template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Vector<RTYPE> Rcpp::union_ (
    const VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
  
```

Definition at line 177 of file setdiff.h.

References `Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get()`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref()`.

Here is the call graph for this function:



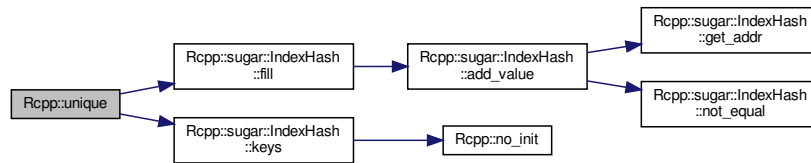
## 5.2.3.391 unique()

```
template<int RTYPE, bool NA, typename T >
Vector<RTYPE> Rcpp::unique (
    const VectorBase< RTYPE, NA, T > & t ) [inline]
```

Definition at line 65 of file unique.h.

References Rcpp::sugar::IndexHash< RTYPE >::fill(), and Rcpp::sugar::IndexHash< RTYPE >::keys().

Here is the call graph for this function:



## 5.2.3.392 unlockBinding()

```
void Rcpp::unlockBinding (
    const std::string & name )
```

unlocks the given binding see ?bindingsIsLocked

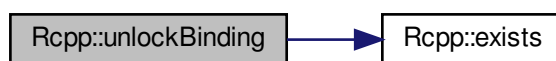
## Exceptions

<i>no_such_binding</i>	if there is no such binding in this environment
------------------------	---

Definition at line 286 of file Environment.h.

References exists().

Here is the call graph for this function:



### 5.2.3.393 unwindProtect()

```
SEXP Rcpp::unwindProtect (
    SEXP(*) (void *data) callback,
    void * data ) [inline]
```

Definition at line 60 of file `unwindProtect.h`.

References `Rcpp::internal::UnwindData::jmpbuf`, and `Rcpp::internal::maybeJump()`.

Here is the call graph for this function:



### 5.2.3.394 update()

```
void Rcpp::update (
    SEXP x )
```

Definition at line 400 of file `Environment.h`.

Referenced by `R_CPP_API_CLASS()`.



### 5.2.3.395 upper\_tri()

```
template<int RTYPE, bool NA, typename T >
sugar::UpperTri<RTYPE, NA, T> Rcpp::upper_tri (
    const Rcpp::MatrixBase< RTYPE, NA, T > & lhs,
    bool diag = false ) [inline]
```

Definition at line 63 of file upper\_tri.h.

References [diag\(\)](#).

Here is the call graph for this function:



### 5.2.3.396 var() [1/4]

```
template<bool NA, typename T >
sugar::Var<CPLXSP,NA,T> Rcpp::var (
    const VectorBase< CPLXSP, NA, T > & t ) [inline]
```

Definition at line 89 of file var.h.

### 5.2.3.397 var() [2/4]

```
template<bool NA, typename T >
sugar::Var<INTSP,NA,T> Rcpp::var (
    const VectorBase< INTSP, NA, T > & t ) [inline]
```

Definition at line 79 of file var.h.

**5.2.3.398 var()** [3/4]

```
template<bool NA, typename T >
sugar::Var<LGLSXP,NA,T> Rcpp::var (
    const VectorBase< LGLSXP, NA, T > & t ) [inline]
```

Definition at line 84 of file var.h.

**5.2.3.399 var()** [4/4]

```
template<bool NA, typename T >
sugar::Var<REALSXP,NA,T> Rcpp::var (
    const VectorBase< REALSXP, NA, T > & t ) [inline]
```

Definition at line 74 of file var.h.

Referenced by Rcpp::sugar::Sd< RTYPE, NA, T >::get().

**5.2.3.400 warning()** [1/12]

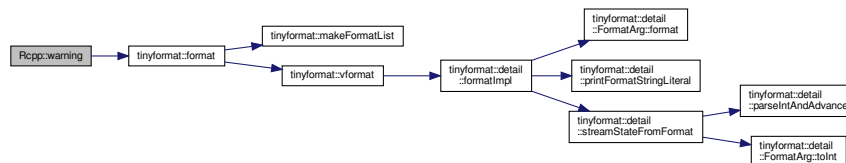
```
template<typename... Args>
void Rcpp::warning (
    const char * fmt,
    Args &&... args ) [inline]
```

Definition at line 46 of file exceptions.h.

References tinyformat::format().

Referenced by r\_cast(), Rcpp::DataFrame\_Impl< StoragePolicy >::set\_type\_after\_push(), and Rcpp::attributes::showWarning().

Here is the call graph for this function:



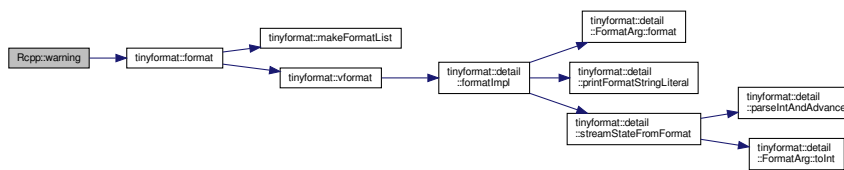
## 5.2.3.401 warning() [2/12]

```
template<typename T1 >
void Rcpp::warning (
    const char * fmt,
    const T1 & arg1 ) [inline]
```

Definition at line 72 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



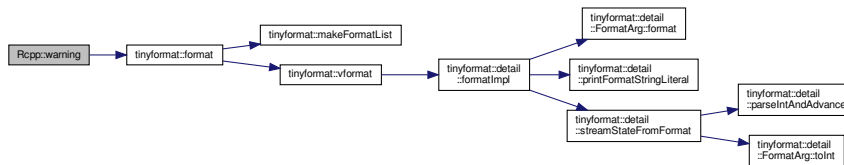
## 5.2.3.402 warning() [3/12]

```
template<typename T1 , typename T2 >
void Rcpp::warning (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2 ) [inline]
```

Definition at line 77 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



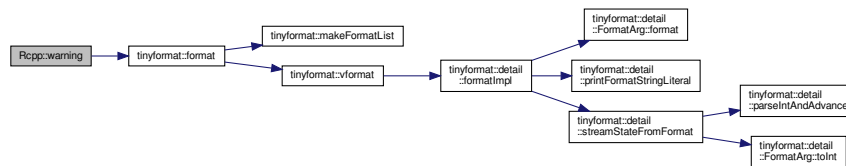
### 5.2.3.403 warning() [4/12]

```
template<typename T1 , typename T2 , typename T3 >
void Rcpp::warning (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3 ) [inline]
```

Definition at line 82 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



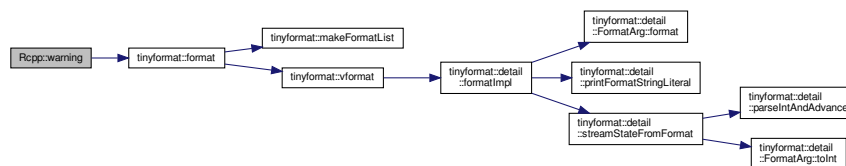
### 5.2.3.404 warning() [5/12]

```
template<typename T1 , typename T2 , typename T3 , typename T4 >
void Rcpp::warning (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4 ) [inline]
```

Definition at line 87 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



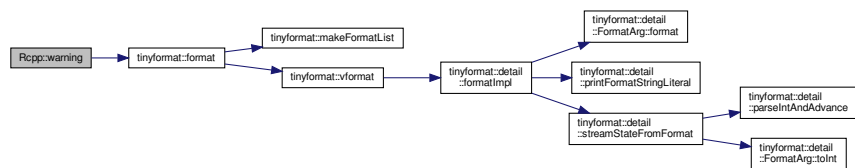
## 5.2.3.405 warning() [6/12]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 >
void Rcpp::warning (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4,
    const T5 & arg5 ) [inline]
```

Definition at line 92 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



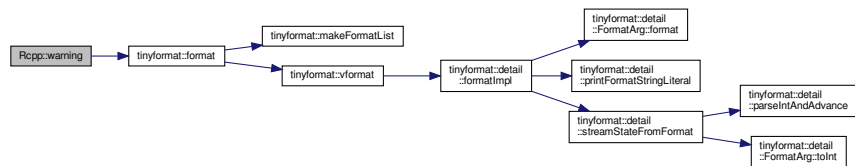
## 5.2.3.406 warning() [7/12]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 >
void Rcpp::warning (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4,
    const T5 & arg5,
    const T6 & arg6 ) [inline]
```

Definition at line 97 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



**5.2.3.407 warning()** [8/12]

```

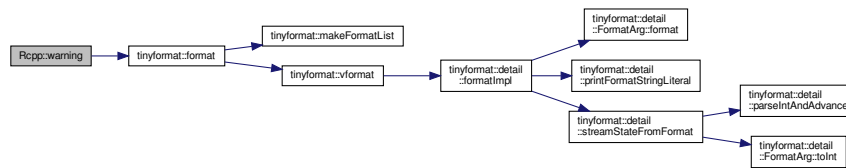
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 >
void Rcpp::warning (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4,
    const T5 & arg5,
    const T6 & arg6,
    const T7 & arg7 ) [inline]

```

Definition at line 102 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:

**5.2.3.408 warning()** [9/12]

```

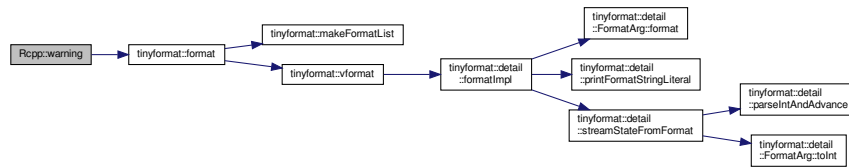
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 >
void Rcpp::warning (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4,
    const T5 & arg5,
    const T6 & arg6,
    const T7 & arg7,
    const T8 & arg8 ) [inline]

```

Definition at line 107 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



### 5.2.3.409 warning() [10/12]

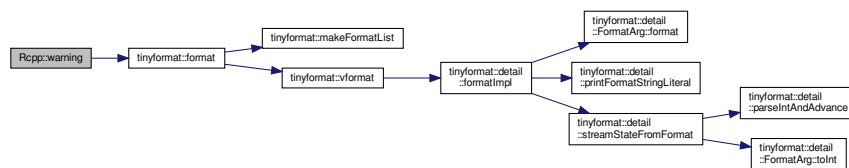
```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 >
void Rcpp::warning (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4,
    const T5 & arg5,
    const T6 & arg6,
    const T7 & arg7,
    const T8 & arg8,
    const T9 & arg9 ) [inline]
  
```

Definition at line 112 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:



**5.2.3.410 warning()** [11/12]

```

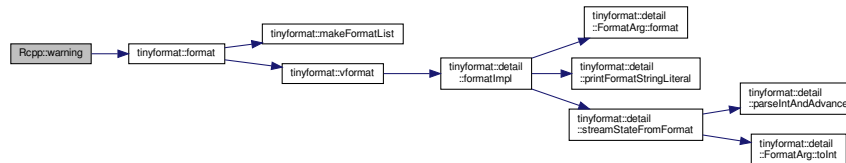
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 >
void Rcpp::warning (
    const char * fmt,
    const T1 & arg1,
    const T2 & arg2,
    const T3 & arg3,
    const T4 & arg4,
    const T5 & arg5,
    const T6 & arg6,
    const T7 & arg7,
    const T8 & arg8,
    const T9 & arg9,
    const T10 & arg10 ) [inline]

```

Definition at line 117 of file exceptions.h.

References `tinyformat::format()`.

Here is the call graph for this function:

**5.2.3.411 warning()** [12/12]

```

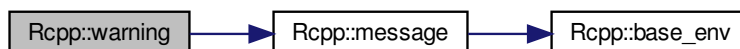
void Rcpp::warning (
    const std::string & message ) [inline]

```

Definition at line 113 of file exceptions.h.

References `message()`.

Here is the call graph for this function:





**5.2.3.412 warningcall()**

```
void Rcpp::warningcall (
    SEXP call,
    const std::string & s ) [inline]
```

Definition at line 29 of file print.h.

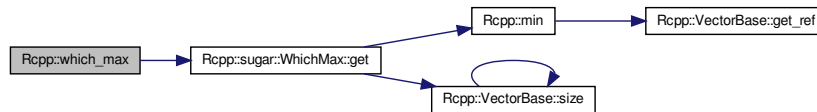
**5.2.3.413 which\_max()**

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::which_max (
    const VectorBase< RTYPE, NA, T > & t )
```

Definition at line 90 of file which\_max.h.

References Rcpp::sugar::WhichMax< RTYPE, NA, T >::get().

Here is the call graph for this function:

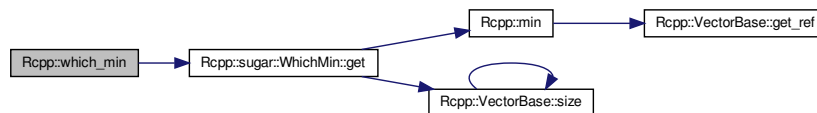
**5.2.3.414 which\_min()**

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::which_min (
    const VectorBase< RTYPE, NA, T > & t )
```

Definition at line 90 of file which\_min.h.

References Rcpp::sugar::WhichMin< RTYPE, NA, T >::get().

Here is the call graph for this function:



### 5.2.3.415 wrap()

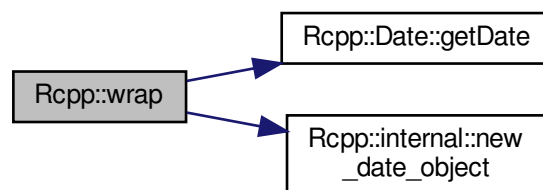
```
template<>
SEXPRcpp::wrap (
    const Date & date ) [inline]
```

Definition at line 38 of file Date.h.

References Rcpp::Date::getDate(), and Rcpp::internal::new\_date\_object().

Referenced by Rcpp::Armor< T >::Armor(), as\_character\_externalptr(), Rcpp::Vector< RTYPE, StoragePolicy >::assign(), Rcpp::Vector< RTYPE, StoragePolicy >::assign\_object(), Rcpp::Vector< RTYPE, StoragePolicy >::assign\_sugar\_expression(), Rcpp::DataFrame\_Impl< StoragePolicy >::DataFrame\_Impl(), CppProperty\_Getter\_Setter< PROP >::get(), CppProperty\_Getter< PROP >::get(), CppProperty\_GetMethod< Class, PROP >::get(), CppProperty\_GetConstMethod< Class, PROP >::get(), CppProperty\_GetPointerMethod< Class, PROP >::get(), CppProperty\_GetMethod\_SetMethod< Class, PROP >::get(), CppProperty\_GetConstMethod\_SetMethod< Class, PROP >::get(), CppProperty\_GetMethod\_SetPointer< Class, PROP >::get(), CppProperty\_GetConstMethod\_SetPointer< Class, PROP >::get(), CppProperty\_GetPointer\_SetMethod< Class, PROP >::get(), CppProperty\_GetPointer\_SetPointer< Class, PROP >::get(), Rcpp::internal::generic\_element\_converter< RTYPE >::get(), Rcpp::sugar::IndexHash< RTYPE >::get\_profile\_data(), Rcpp::internal::grow\_dispatch(), Rcpp::Dimension::operator SEXP(), Rcpp::sugar::Sign< RTYPE, NA, T >::operator SEXP(), Rcpp::ListOf< T >::operator SEXP(), Rcpp::internal::simple\_name\_proxy< RTYPE, StoragePolicy >::operator SEXP(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_< RTYPE, RHS\_NA, RHS\_T >::operator SEXP(), Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator=(), Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator=(), Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator=(), Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::operator=(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator=(), Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=(), Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator=(), Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::operator=(), Rcpp::internal::generic\_name\_proxy< RTYPE, StoragePolicy >::operator=(), Rcpp::ChildVector< T >::operator=(), Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator=(), Rcpp::Armor< T >::operator=(), Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::operator[](), PiLeibniz(), Rcpp::internal::range\_wrap\_dispatch\_impl\_pair(), and RCPP\_API\_CLASS().

Here is the call graph for this function:



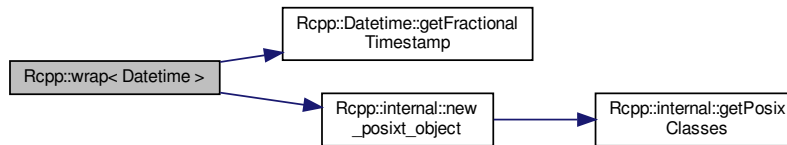
**5.2.3.416 wrap< Datetime >()**

```
template<>
SEXPR Rcpp::wrap< Datetime > (
    const Datetime & date ) [inline]
```

Definition at line 44 of file Datetime.h.

References Rcpp::Datetime::getFractionalTimestamp(), and Rcpp::internal::new\_posixt\_object().

Here is the call graph for this function:

**5.2.3.417 wrap< Rcpp::Date >()**

```
template<>
SEXPR Rcpp::wrap< Rcpp::Date > (
    const Rcpp::Date & date )
```

**5.2.3.418 wrap< Rcpp::Datetime >()**

```
template<>
SEXPR Rcpp::wrap< Rcpp::Datetime > (
    const Rcpp::Datetime & dt )
```

**5.2.3.419 wrap< Rcpp::String >()**

```
template<>
SEXPR Rcpp::wrap< Rcpp::String > (
    const Rcpp::String & object ) [inline]
```

Definition at line 711 of file String.h.

References RCPP\_STRING\_DEBUG.

### 5.2.3.420 wrap\_extra\_steps()

```
template<typename T >
SEXP Rcpp::wrap_extra_steps (
    SEXP x )
```

Definition at line 28 of file wrap\_extra\_steps.h.

### 5.2.3.421 wrap\_extra\_steps< Rcpp::Date >()

```
template<>
SEXP Rcpp::wrap_extra_steps< Rcpp::Date > (
    SEXP x ) [inline]
```

Definition at line 151 of file Date.h.

### 5.2.3.422 wrap\_extra\_steps< Rcpp::Datetime >()

```
template<>
SEXP Rcpp::wrap_extra_steps< Rcpp::Datetime > (
    SEXP ) [inline]
```

Definition at line 38 of file Datetime.h.

References Rcpp::internal::getPosixClasses().

Here is the call graph for this function:



## 5.2.4 Variable Documentation

### 5.2.4.1 `_`

```
internal::NamedPlaceholder Rcpp::_ [static]
```

Definition at line 64 of file Named.h.

Referenced by `Rcpp::attributes::checkRSignature()`, `Rcpp::exception::copy_stack_trace_to_r()`, `Rcpp::Module::invoke()`, `new_env()`, `shush_about_NA()`, and `sourceCppContext()`.

### 5.2.4.2 `BindingPolicy< Environment_Impl< StoragePolicy > >`

```
public Rcpp::BindingPolicy< Environment_Impl< StoragePolicy > >
```

#### Initial value:

```
{
  private:
    inline SEXP as_environment(SEXP x){
      if( Rf_isEnvironment(x) ) return x ;
      SEXP asEnvironmentSym = Rf_install("as.environment");
      try {
        Shield<SEXP> call(Rf_lang2(asEnvironmentSym, x));
        return Rcpp_fast_eval(call, R_GlobalEnv);
      } catch( const eval_error& ex) {
        const char* fmt = "Cannot convert object to an environment: "
                          "[type=%s; target=ENVSXP].";
        throw not_compatible(fmt, Rf_type2char(TYPEOF(x)));
      }
    }
  public:
    Environment_Impl(){
      Storage::set__(R_GlobalEnv) ;
    }
}
```

Definition at line 29 of file Environment.h.

### 5.2.4.3 `current_scope`

```
Module* Rcpp::current_scope [static]
```

Definition at line 224 of file module.cpp.

Referenced by `getCurrentScope()`, and `setCurrentScope()`.

#### 5.2.4.4 DottedPairImpl< DottedPair\_Impl< StoragePolicy > >

```
public public Rcpp::DottedPairImpl< DottedPair_Impl< StoragePolicy > >
```

##### Initial value:

```
{
public:
    DottedPair_Impl() {}
    DottedPair_Impl(SEXP x) {
        Storage::set__(x) ;
    }
    void update(SEXP) {}
}
```

Definition at line 29 of file DottedPair.h.

#### 5.2.4.5 DottedPairImpl< Formula\_Impl< StoragePolicy > >

```
public public Rcpp::DottedPairImpl< Formula_Impl< StoragePolicy > >
```

##### Initial value:

```
{
    public:
        Formula_Impl() {}
}
```

Definition at line 32 of file Formula.h.

#### 5.2.4.6 DottedPairImpl< Language\_Impl< StoragePolicy > >

```
public public Rcpp::DottedPairImpl< Language_Impl< StoragePolicy > >
```

##### Initial value:

```
{
    public:
        typedef typename DottedPairProxyPolicy<Language_Impl>::DottedPairProxy Proxy
}
```

Definition at line 33 of file Language.h.

#### 5.2.4.7 DottedPairImpl< Pairlist\_Impl< StoragePolicy > >

```
public public Rcpp::DottedPairImpl< Pairlist_Impl< StoragePolicy > >
```

##### Initial value:

```
{
    public:
        typedef typename DottedPairProxyPolicy<Pairlist_Impl>::DottedPairProxy Proxy
}
```

Definition at line 33 of file Pairlist.h.

#### 5.2.4.8 DottedPairProxyPolicy< DottedPair\_Impl< StoragePolicy > >

```
public Rcpp::DottedPairProxyPolicy< DottedPair_Impl< StoragePolicy > >
```

Definition at line 28 of file DottedPair.h.

#### 5.2.4.9 DottedPairProxyPolicy< Formula\_Impl< StoragePolicy > >

```
public Rcpp::DottedPairProxyPolicy< Formula_Impl< StoragePolicy > >
```

Definition at line 31 of file Formula.h.

#### 5.2.4.10 DottedPairProxyPolicy< Language\_Impl< StoragePolicy > >

```
public Rcpp::DottedPairProxyPolicy< Language_Impl< StoragePolicy > >
```

Definition at line 32 of file Language.h.

#### 5.2.4.11 DottedPairProxyPolicy< Pairlist\_Impl< StoragePolicy > >

```
public Rcpp::DottedPairProxyPolicy< Pairlist_Impl< StoragePolicy > >
```

Definition at line 32 of file Pairlist.h.

#### 5.2.4.12 DottedPairProxyPolicy< StretchyList\_Impl< StoragePolicy > >

```
public Rcpp::DottedPairProxyPolicy< StretchyList_Impl< StoragePolicy > >
```

**Initial value:**

```
{  
    public:  
        typedef typename DottedPairProxyPolicy<StretchyList_Impl>::DottedPairProxy Proxy
```

Definition at line 34 of file StretchyList.h.

### 5.2.4.13 FieldProxyPolicy< Reference\_Impl< StoragePolicy > >

```
public Rcpp::FieldProxyPolicy< Reference_Impl< StoragePolicy > >
```

#### Initial value:

```
{
  public:
    Reference_Impl() {}

    Reference_Impl(SEXP x) {
      Storage::set__(x) ;
    }
    Reference_Impl& operator=( SEXP other ) {
      Storage::set__(other) ;
      return *this ;
    }

    Reference_Impl( const std::string& klass ) {
      SEXP newSym = Rf_install("new");
      Shield<SEXP> str(Rf_mkString(klass.c_str()));
      Shield<SEXP> call(Rf_lang2(newSym, str));
      Storage::set__( Rcpp_fast_eval( call , Rcpp::internal::get_Rcpp_namespace() ) );
    }
    void update( SEXP x){
      if( ! ::Rf_isS4(x) ) throw not_reference();
    }
}
```

Definition at line 31 of file Reference.h.

### 5.2.4.14 gmt

```
const char Rcpp::gmt[] = "GMT" [static]
```

Definition at line 358 of file date.cpp.

Referenced by gmtload().

### 5.2.4.15 gmt\_is\_set

```
int Rcpp::gmt_is_set [static]
```

Definition at line 429 of file date.cpp.

Referenced by gmtsub().

### 5.2.4.16 gmtmem

```
struct state Rcpp::gmtmem [static]
```

Definition at line 429 of file date.cpp.



### 5.2.4.17 mon\_lengths

```
const int Rcpp::mon_lengths[2][MONSPERYEAR] [static]
```

**Initial value:**

```
= {  
    { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 },  
    { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }  
}
```

Definition at line 420 of file date.cpp.

Referenced by timesub(), and transtime().

### 5.2.4.18 NA

```
Na_Proxy Rcpp::NA [static]
```

Definition at line 52 of file Na\_Proxy.h.

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::import\_sugar\_expression(), Rcpp::algorithm::max(), Rcpp::algorithm::min(), Rcpp::algorithm::helpers::rtype< T >::NA(), operator!=(), operator<(), operator<=(), operator==(), operator>(), operator>=(), Rcpp::algorithm::prod(), sapply(), shush\_about\_NA(), and Rcpp::algorithm::sum().

### 5.2.4.19 Rcerr

```
Rostream< false > & Rcpp::Rcerr = Rcpp_cerr_get() [static]
```

Definition at line 90 of file Rstreambuf.h.

### 5.2.4.20 Rcout

```
Rostream< true > & Rcpp::Rcout = Rcpp_cout_get() [static]
```

Definition at line 89 of file Rstreambuf.h.

Referenced by compileAttributes(), and Rcpp::attributes::CppExportsGenerator::doWriteFunctions().

#### 5.2.4.21 Rcpp\_precious

```
SEXP Rcpp::Rcpp_precious = R_NilValue [static]
```

Definition at line 92 of file barrier.cpp.

#### 5.2.4.22 time\_t\_max

```
time_t const Rcpp::time_t_max = MAXVAL(time_t, TYPE_BIT(time_t)) [static]
```

Definition at line 162 of file date.cpp.

Referenced by `increment_overflow_time()`.

#### 5.2.4.23 time\_t\_min

```
time_t const Rcpp::time_t_min = MINVAL(time_t, TYPE_BIT(time_t)) [static]
```

Definition at line 161 of file date.cpp.

Referenced by `increment_overflow_time()`.

#### 5.2.4.24 tm

```
struct tm Rcpp::tm [static]
```

Definition at line 429 of file date.cpp.

Referenced by `Rcpp::Datetime::format()`, `Rcpp::Date::format()`, `gmtime_()`, `gmtsub()`, and `mktime00()`.

#### 5.2.4.25 year\_lengths

```
const int Rcpp::year_lengths[2] [static]
```

**Initial value:**

```
= {  
    DAYSPERNYEAR, DAYSPERLYEAR  
}
```

Definition at line 425 of file date.cpp.

Referenced by `timesub()`, and `tzparse()`.

## 5.3 Rcpp::algorithm Namespace Reference

### Namespaces

- [helpers](#)

### Functions

- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↵`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type` `sum`  
`(InputIterator begin, InputIterator end)`
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↵`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type` `sum`  
`nona` `(InputIterator begin, InputIterator end)`
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↵`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type` `prod`  
`(InputIterator begin, InputIterator end)`
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↵`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type` `prod`  
`nona` `(InputIterator begin, InputIterator end)`
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↵`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type` `max`  
`(InputIterator begin, InputIterator end)`
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↵`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type` `max`  
`nona` `(InputIterator begin, InputIterator end)`
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↵`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type` `min`  
`(InputIterator begin, InputIterator end)`
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >↵`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type` `min`  
`nona` `(InputIterator begin, InputIterator end)`
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value`  
`&&traits::same_type< typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >↵`  
`::type, double >::value, double >::type` `mean` `(InputIterator begin, InputIterator end)`
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value`  
`&&traits::same_type< typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >↵`  
`::type, int >::value, double >::type` `mean` `(InputIterator begin, InputIterator end)`
- `template<typename InputIterator, typename OutputIterator >`  
`void` `log` `(InputIterator begin, InputIterator end, OutputIterator out)`
- `template<typename InputIterator, typename OutputIterator >`  
`void` `exp` `(InputIterator begin, InputIterator end, OutputIterator out)`
- `template<typename InputIterator, typename OutputIterator >`  
`void` `sqrt` `(InputIterator begin, InputIterator end, OutputIterator out)`

## 5.3.1 Function Documentation

### 5.3.1.1 `exp()`

```
template<typename InputIterator , typename OutputIterator >
void Rcpp::algorithm::exp (
    InputIterator begin,
    InputIterator end,
    OutputIterator out )
```

Definition at line 474 of file `algorithm.h`.

Referenced by `Rcpp::internal::complex__exp()`, `Rcpp::stats::d_exp_0()`, `Rcpp::stats::dlnorm_0()`, `Rcpp::stats::dlnorm_1()`, `Rcpp::stats::dlogis_0()`, `Rcpp::stats::dlogis_1()`, `Rcpp::stats::dnorm_0()`, `Rcpp::stats::dnorm_1()`, `Rcpp::stats::dweibull_1()`, `Rcpp::stats::LNormGenerator::operator()`, `Rcpp::stats::LNormGenerator_1::operator()`, `Rcpp::stats::LNormGenerator_0::operator()`, `Rcpp::algorithm::helpers::exp::operator()`, `Rcpp::stats::p_exp_0()`, `Rcpp::stats::plogis_0()`, `Rcpp::stats::plogis_1()`, `Rcpp::stats::pweibull_1()`, `Rcpp::stats::qlnorm_0()`, `Rcpp::stats::qlnorm_1()`, `Rcpp::stats::qlogis_0()`, `Rcpp::stats::qlogis_1()`, and `Rcpp::rlnorm()`.

### 5.3.1.2 `log()`

```
template<typename InputIterator , typename OutputIterator >
void Rcpp::algorithm::log (
    InputIterator begin,
    InputIterator end,
    OutputIterator out )
```

Definition at line 469 of file `algorithm.h`.

Referenced by `Rcpp::internal::complex__asin()`, `Rcpp::internal::complex__atan()`, `Rcpp::internal::complex__log()`, `Rcpp::dexp()`, `Rcpp::stats::dgamma_1()`, `Rcpp::stats::dlnorm_0()`, `Rcpp::stats::dlnorm_1()`, `Rcpp::stats::dlogis_0()`, `Rcpp::stats::dlogis_1()`, `Rcpp::stats::dweibull_1()`, `Rcpp::stats::LogisGenerator::operator()`, `Rcpp::stats::LogisGenerator_1::operator()`, `Rcpp::stats::LogisGenerator_0::operator()`, `Rcpp::stats::WeibullGenerator::operator()`, `Rcpp::stats::WeibullGenerator__scale1::operator()`, `Rcpp::algorithm::helpers::log::operator()`, `Rcpp::stats::p_exp_0()`, `Rcpp::pexp()`, `Rcpp::stats::plnorm_0()`, `Rcpp::stats::plnorm_1()`, `Rcpp::stats::pweibull_1()`, `Rcpp::qexp()`, `Rcpp::stats::qlogis_0()`, and `Rcpp::stats::qlogis_1()`.

### 5.3.1.3 max()

```
template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >↔
::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >↔
::value_type >::type >::type Rcpp::algorithm::max (
    InputIterator begin,
    InputIterator end )
```

Definition at line 324 of file algorithm.h.

References Rcpp::NA.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::check\_indices(), max\_nona(), and Rcpp::operator<<().

### 5.3.1.4 max\_nona()

```
template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >↔
::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >↔
::value_type >::type >::type Rcpp::algorithm::max_nona (
    InputIterator begin,
    InputIterator end )
```

Definition at line 349 of file algorithm.h.

References max().

Here is the call graph for this function:



### 5.3.1.5 mean() [1/2]

```
template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >↵
::value_type >::value && traits::same_type< typename helpers::ctype< typename std::iterator_↵
traits< InputIterator >::value_type >::type, double >::value, double >::type Rcpp::algorithm↵
::mean (
    InputIterator begin,
    InputIterator end )
```

Definition at line 417 of file algorithm.h.

References Rcpp::algorithm::helpers::rtype< T >::NA().

Here is the call graph for this function:



### 5.3.1.6 mean() [2/2]

```
template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >↵
::value_type >::value && traits::same_type< typename helpers::ctype< typename std::iterator_↵
traits< InputIterator >::value_type >::type, int >::value, double >::type Rcpp::algorithm::mean
(
    InputIterator begin,
    InputIterator end )
```

Definition at line 444 of file algorithm.h.

References Rcpp::algorithm::helpers::rtype< T >::NA().

Here is the call graph for this function:



### 5.3.1.7 min()

```
template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >↔
::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >↔
::value_type >::type >::type Rcpp::algorithm::min (
    InputIterator begin,
    InputIterator end )
```

Definition at line 370 of file algorithm.h.

References Rcpp::NA.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::fill\_diag\_dispatch(), tinyformat::detail::formatTruncated(), min\_nona(), Rcpp::operator<<(), and PiLeibniz().

### 5.3.1.8 min\_nona()

```
template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >↔
::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >↔
::value_type >::type >::type Rcpp::algorithm::min_nona (
    InputIterator begin,
    InputIterator end )
```

Definition at line 395 of file algorithm.h.

References min().

Here is the call graph for this function:



### 5.3.1.9 prod()

```
template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >↵
::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >↵
::value_type >::type >::type Rcpp::algorithm::prod (
    InputIterator begin,
    InputIterator end )
```

Definition at line 278 of file algorithm.h.

References Rcpp::NA.

### 5.3.1.10 prod\_nona()

```
template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >↵
::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >↵
::value_type >::type >::type Rcpp::algorithm::prod_nona (
    InputIterator begin,
    InputIterator end )
```

Definition at line 303 of file algorithm.h.

### 5.3.1.11 sqrt()

```
template<typename InputIterator , typename OutputIterator >
void Rcpp::algorithm::sqrt (
    InputIterator begin,
    InputIterator end,
    OutputIterator out )
```

Definition at line 479 of file algorithm.h.

Referenced by Rcpp::internal::complex\_asin(), Rcpp::internal::complex\_Mod(), Rcpp::internal::complex\_sqrt(), fastLm(), Rcpp::sugar::Sd< RTYPE, NA, T >::get(), Rcpp::stats::TGenerator::operator>(), Rcpp::algorithm::helpers↵
::sqrt::operator>(), and piSugar().



### 5.3.1.12 sum()

```
template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >↔
::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >↔
::value_type >::type >::type Rcpp::algorithm::sum (
    InputIterator begin,
    InputIterator end )
```

Definition at line 232 of file algorithm.h.

References Rcpp::NA.

### 5.3.1.13 sum\_nona()

```
template<typename InputIterator >
traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >↔
::value_type >::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >↔
::value_type >::type >::type Rcpp::algorithm::sum_nona (
    InputIterator begin,
    InputIterator end )
```

Definition at line 257 of file algorithm.h.

## 5.4 Rcpp::algorithm::helpers Namespace Reference

### Classes

- struct [CTYPE\\_CHAR](#)
- struct [CTYPE\\_SHORT](#)
- struct [CTYPE\\_INT](#)
- struct [CTYPE\\_LONG](#)
- struct [CTYPE\\_FLOAT](#)
- struct [CTYPE\\_DOUBLE](#)
- struct [CTYPE\\_LONG\\_DOUBLE](#)
- struct [CTYPE\\_STRING](#)
- struct [CTYPE\\_UNSIGNED\\_CHAR](#)
- struct [CTYPE\\_UNSIGNED\\_SHORT](#)
- struct [CTYPE\\_UNSIGNED\\_INT](#)
- struct [CTYPE\\_UNSIGNED\\_LONG](#)
- struct [CTYPE\\_UNKNOWN](#)
- struct [ctype\\_helper](#)
- struct [ctype\\_helper< sizeof\(CTYPE\\_CHAR\) >](#)
- struct [ctype\\_helper< sizeof\(CTYPE\\_SHORT\) >](#)
- struct [ctype\\_helper< sizeof\(CTYPE\\_INT\) >](#)
- struct [ctype\\_helper< sizeof\(CTYPE\\_LONG\) >](#)

- struct [ctype\\_helper< sizeof\(CTYPE\\_FLOAT\) >](#)
- struct [ctype\\_helper< sizeof\(CTYPE\\_DOUBLE\) >](#)
- struct [ctype\\_helper< sizeof\(CTYPE\\_LONG\\_DOUBLE\) >](#)
- struct [ctype\\_helper< sizeof\(CTYPE\\_STRING\) >](#)
- struct [ctype\\_helper< sizeof\(CTYPE\\_UNSIGNED\\_CHAR\) >](#)
- struct [ctype\\_helper< sizeof\(CTYPE\\_UNSIGNED\\_SHORT\) >](#)
- struct [ctype\\_helper< sizeof\(CTYPE\\_UNSIGNED\\_INT\) >](#)
- struct [ctype\\_helper< sizeof\(CTYPE\\_UNSIGNED\\_LONG\) >](#)
- struct [ctype](#)
- struct [decays\\_to\\_ctype](#)
- struct [rtype\\_helper](#)
- struct [rtype\\_helper< double >](#)
- struct [rtype\\_helper< int >](#)
- struct [rtype](#)
- struct [log](#)
- struct [exp](#)
- struct [sqrt](#)

## 5.5 Rcpp::attributes Namespace Reference

### Classes

- class [FileInfo](#)
- class [Type](#)
- class [Argument](#)
- class [Function](#)
- class [Param](#)
- class [Attribute](#)
- class [SourceFileAttributes](#)
- class [CommentState](#)
- class [SourceFileAttributesParser](#)
- class [ExportsGenerator](#)
- class [CppExportsGenerator](#)
- class [CppExportsIncludeGenerator](#)
- class [CppPackageIncludeGenerator](#)
- class [REExportsGenerator](#)
- class [ExportsGenerators](#)

### Functions

- bool [removeFile](#) (const std::string &path)
- void [createDirectory](#) (const std::string &path)
- bool [isWhitespace](#) (char ch)
- void [trimWhitespace](#) (std::string \*pStr)
- void [stripTrailingLineComments](#) (std::string \*pStr)
- void [stripQuotes](#) (std::string \*pStr)
- bool [isQuoted](#) (const std::string &str)

- bool [endsWith](#) (const std::string &str, const std::string &suffix)
- void [showWarning](#) (const std::string &msg)
- bool [isRoxygenCpp](#) (const std::string &str)
- std::ostream & [operator<<](#) (std::ostream &os, const [Type](#) &type)
- std::ostream & [operator<<](#) (std::ostream &os, const [Argument](#) &argument)
- std::ostream & [operator<<](#) (std::ostream &os, const [Function](#) &function)
- std::ostream & [operator<<](#) (std::ostream &os, const [Param](#) &param)
- std::ostream & [operator<<](#) (std::ostream &os, const [Attribute](#) &attribute)
- std::string [generateRArgList](#) (const [Function](#) &function)
- bool [checkRSignature](#) (const [Function](#) &function, std::string args)
- void [initializeGlobals](#) (std::ostream &ostr)
- void [generateCpp](#) (std::ostream &ostr, const [SourceFileAttributes](#) &attributes, bool includePrototype, bool cpp↔  
Interface, const std::string &contextId)
- void [printArgument](#) (std::ostream &os, const [Argument](#) &argument, bool printDefault=true)
- void [printFunction](#) (std::ostream &os, const [Function](#) &function, bool printArgDefaults=true)

## Variables

- const char \*const [kWhitespaceChars](#) = "\n\r\t\v"
- const char \*const [kExportAttribute](#) = "export"
- const char \*const [kExportName](#) = "name"
- const char \*const [kExportRng](#) = "rng"
- const char \*const [kExportInvisible](#) = "invisible"
- const char \*const [kExportSignature](#) = "signature"
- const char \*const [kInitAttribute](#) = "init"
- const char \*const [kDependsAttribute](#) = "depends"
- const char \*const [kPluginsAttribute](#) = "plugins"
- const char \*const [kInterfacesAttribute](#) = "interfaces"
- const char \*const [kInterfaceR](#) = "r"
- const char \*const [kInterfaceCpp](#) = "cpp"
- const char \*const [kParamValueFalse](#) = "false"
- const char \*const [kParamValueTrue](#) = "true"
- const char \*const [kParamValueFALSE](#) = "FALSE"
- const char \*const [kParamValueTRUE](#) = "TRUE"
- const char \*const [kParamBlockStart](#) = "{,"
- const char \*const [kParamBlockEnd](#) = "}"

### 5.5.1 Function Documentation



### 5.5.1.3 endsWith()

```
bool Rcpp::attributes::endsWith (
    const std::string & str,
    const std::string & suffix )
```

Definition at line 3109 of file attributes.cpp.

Referenced by compileAttributes().

### 5.5.1.4 generateCpp()

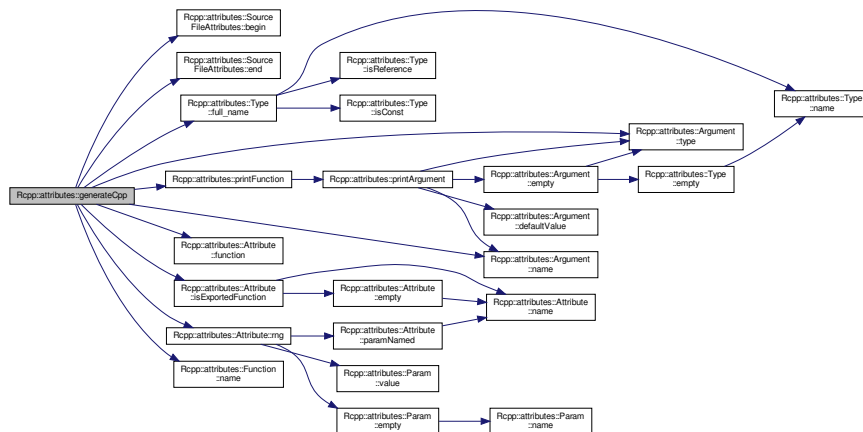
```
void Rcpp::attributes::generateCpp (
    std::ostream & ostr,
    const SourceFileAttributes & attributes,
    bool includePrototype,
    bool cppInterface,
    const std::string & contextId )
```

Definition at line 2843 of file attributes.cpp.

References Rcpp::attributes::SourceFileAttributes::begin(), Rcpp::attributes::SourceFileAttributes::end(), Rcpp::attributes::Type::full\_name(), Rcpp::attributes::Attribute::function(), Rcpp::attributes::Attribute::isExportedFunction(), Rcpp::attributes::Argument::name(), Rcpp::attributes::Function::name(), printFunction(), Rcpp::attributes::Attribute::rng(), and Rcpp::attributes::Argument::type().

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions().

Here is the call graph for this function:



### 5.5.1.5 generateRArgList()

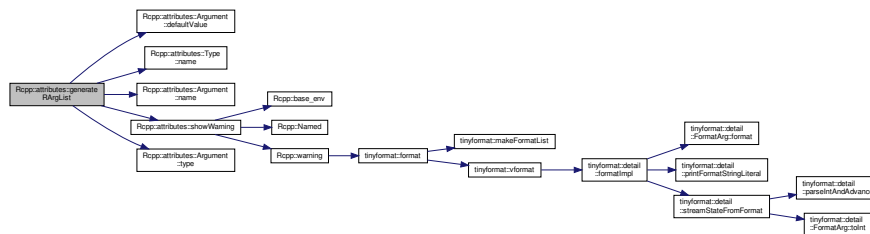
```
std::string Rcpp::attributes::generateRArgList (
    const Function & function )
```

Definition at line 2778 of file attributes.cpp.

References `Rcpp::attributes::Argument::defaultValue()`, `Rcpp::attributes::Type::name()`, `Rcpp::attributes::Argument::name()`, `showWarning()`, and `Rcpp::attributes::Argument::type()`.

Referenced by `Rcpp::attributes::REExportsGenerator::doWriteFunctions()`.

Here is the call graph for this function:



### 5.5.1.6 initializeGlobals()

```
void Rcpp::attributes::initializeGlobals (
    std::ostream & ostr )
```

Definition at line 2832 of file attributes.cpp.

Referenced by `Rcpp::attributes::CppExportsGenerator::commit()`.

### 5.5.1.7 isQuoted()

```
bool Rcpp::attributes::isQuoted (
    const std::string & str )
```

Definition at line 3101 of file attributes.cpp.

### 5.5.1.8 isRoxygenCpp()

```
bool Rcpp::attributes::isRoxygenCpp (
    const std::string & str )
```

Definition at line 3121 of file attributes.cpp.

References kWhitespaceChars.

Referred by stripTrailingLineComments().

### 5.5.1.9 isWhitespace()

```
bool Rcpp::attributes::isWhitespace (
    char ch )
```

Definition at line 3025 of file attributes.cpp.

References kWhitespaceChars.

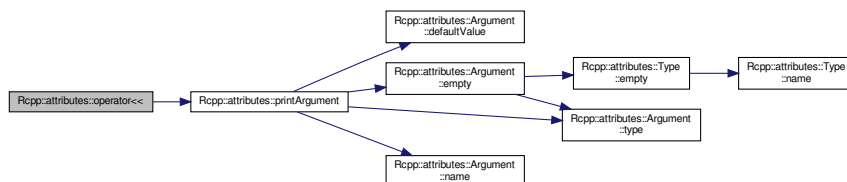
### 5.5.1.10 operator<<() [1/5]

```
std::ostream & Rcpp::attributes::operator<< (
    std::ostream & os,
    const Argument & argument )
```

Definition at line 1119 of file attributes.cpp.

References printArgument().

Here is the call graph for this function:



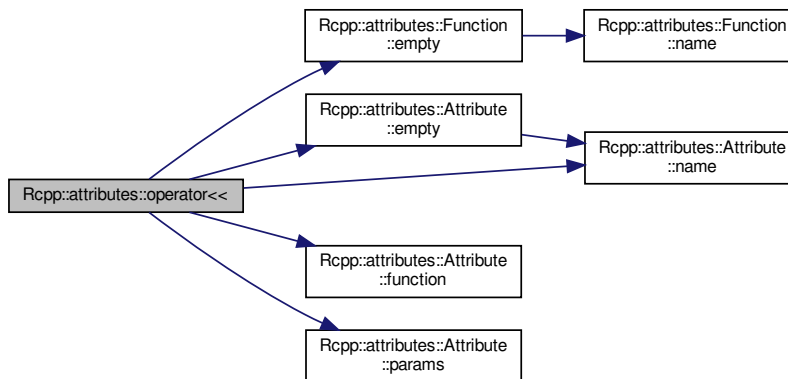
### 5.5.1.11 operator<<() [2/5]

```
std::ostream & Rcpp::attributes::operator<< (
    std::ostream & os,
    const Attribute & attribute )
```

Definition at line 1163 of file attributes.cpp.

References Rcpp::attributes::Function::empty(), Rcpp::attributes::Attribute::empty(), Rcpp::attributes::Attribute::function(), Rcpp::attributes::Attribute::name(), and Rcpp::attributes::Attribute::params().

Here is the call graph for this function:



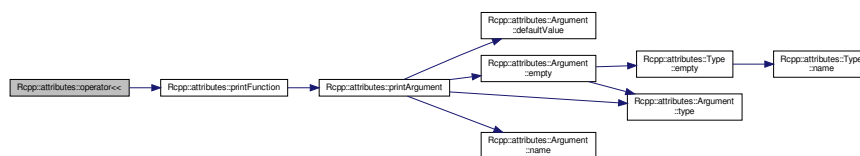
### 5.5.1.12 operator<<() [3/5]

```
std::ostream & Rcpp::attributes::operator<< (
    std::ostream & os,
    const Function & function )
```

Definition at line 1147 of file attributes.cpp.

References `printFunction()`.

Here is the call graph for this function:





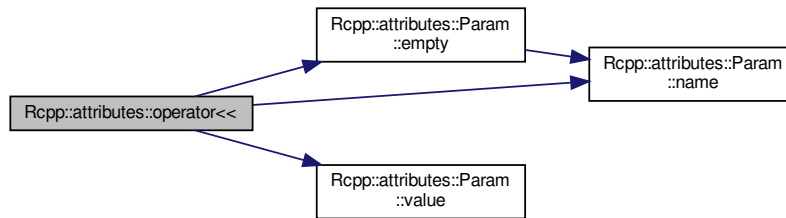
## 5.5.1.13 operator&lt;&lt;() [4/5]

```
std::ostream & Rcpp::attributes::operator<< (
    std::ostream & os,
    const Param & param )
```

Definition at line 1153 of file attributes.cpp.

References Rcpp::attributes::Param::empty(), Rcpp::attributes::Param::name(), and Rcpp::attributes::Param::value().

Here is the call graph for this function:



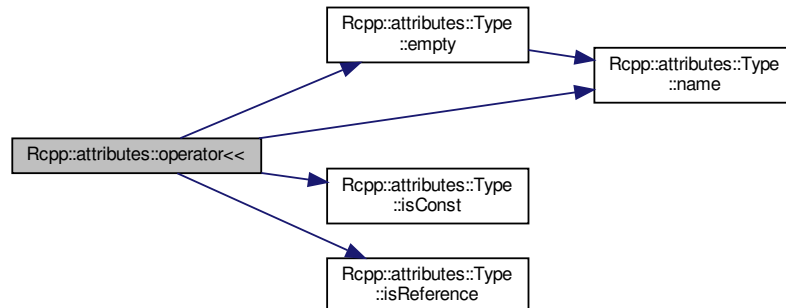
## 5.5.1.14 operator&lt;&lt;() [5/5]

```
std::ostream & Rcpp::attributes::operator<< (
    std::ostream & os,
    const Type & type )
```

Definition at line 1092 of file attributes.cpp.

References Rcpp::attributes::Type::empty(), Rcpp::attributes::Type::isConst(), Rcpp::attributes::Type::isReference(), and Rcpp::attributes::Type::name().

Here is the call graph for this function:



### 5.5.1.15 printArgument()

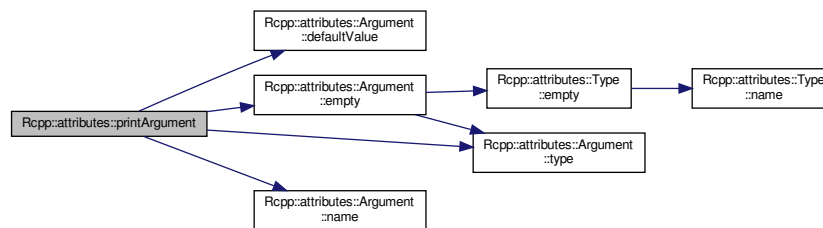
```
void Rcpp::attributes::printArgument (
    std::ostream & os,
    const Argument & argument,
    bool printDefault = true )
```

Definition at line 1104 of file attributes.cpp.

References `Rcpp::attributes::Argument::defaultValue()`, `Rcpp::attributes::Argument::empty()`, `Rcpp::attributes::Argument::name()`, and `Rcpp::attributes::Argument::type()`.

Referenced by operator<<(), and `printFunction()`.

Here is the call graph for this function:



### 5.5.1.16 printFunction()

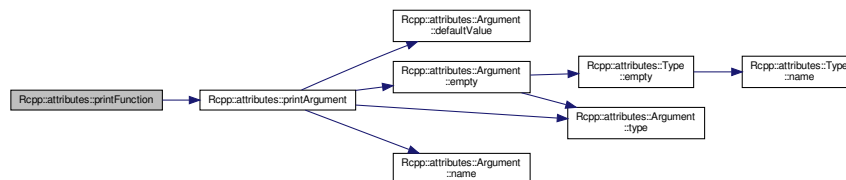
```
void Rcpp::attributes::printFunction (
    std::ostream & os,
    const Function & function,
    bool printArgDefaults = true )
```

Definition at line 1125 of file attributes.cpp.

References `printArgument()`.

Referenced by `generateCpp()`, operator<<(), and `Rcpp::attributes::CppExportsGenerator::writeEnd()`.

Here is the call graph for this function:



## 5.5.1.17 removeFile()

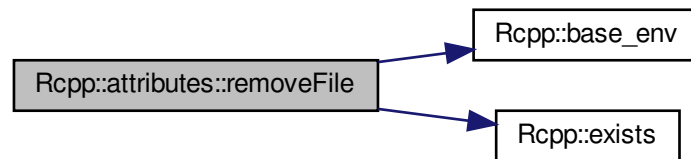
```
bool Rcpp::attributes::removeFile (
    const std::string & path )
```

Definition at line 3002 of file attributes.cpp.

References Rcpp::base\_env(), and Rcpp::exists().

Referenced by Rcpp::attributes::ExportsGenerator::remove().

Here is the call graph for this function:



## 5.5.1.18 showWarning()

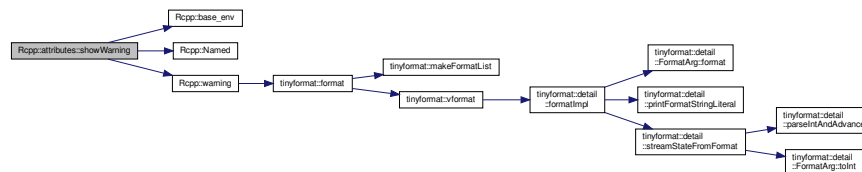
```
void Rcpp::attributes::showWarning (
    const std::string & msg )
```

Definition at line 3116 of file attributes.cpp.

References Rcpp::base\_env(), Rcpp::Named(), and Rcpp::warning().

Referenced by compileAttributes(), generateRArgList(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

Here is the call graph for this function:



### 5.5.1.19 stripQuotes()

```
void Rcpp::attributes::stripQuotes (
    std::string * pStr )
```

Definition at line 3092 of file attributes.cpp.

Referenced by Rcpp::attributes::Param::Param().

### 5.5.1.20 stripTrailingLineComments()

```
void Rcpp::attributes::stripTrailingLineComments (
    std::string * pStr )
```

Definition at line 3031 of file attributes.cpp.

References isRoxygenCpp(), and kWhitespaceChars.

Here is the call graph for this function:



### 5.5.1.21 trimWhitespace()

```
void Rcpp::attributes::trimWhitespace (
    std::string * pStr )
```

Definition at line 3075 of file attributes.cpp.

References kWhitespaceChars.

Referenced by Rcpp::attributes::Attribute::customRSignature(), and Rcpp::attributes::Param::Param().

## 5.5.2 Variable Documentation

### 5.5.2.1 kDependsAttribute

```
const char* const Rcpp::attributes::kDependsAttribute = "depends"
```

Definition at line 159 of file attributes.cpp.

Referenced by compileAttributes().

### 5.5.2.2 kExportAttribute

```
const char* const Rcpp::attributes::kExportAttribute = "export"
```

Definition at line 153 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::isExportedFunction().

### 5.5.2.3 kExportInvisible

```
const char* const Rcpp::attributes::kExportInvisible = "invisible"
```

Definition at line 156 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::invisible().

### 5.5.2.4 kExportName

```
const char* const Rcpp::attributes::kExportName = "name"
```

Definition at line 154 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::exportedName().

### 5.5.2.5 kExportRng

```
const char* const Rcpp::attributes::kExportRng = "rng"
```

Definition at line 155 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::rng().

### 5.5.2.6 kExportSignature

```
const char* const Rcpp::attributes::kExportSignature = "signature"
```

Definition at line 157 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::customRSignature(), and Rcpp::attributes::REExportsGenerator::doWriteFunctions().

### 5.5.2.7 kInitAttribute

```
const char* const Rcpp::attributes::kInitAttribute = "init"
```

Definition at line 158 of file attributes.cpp.

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions().

### 5.5.2.8 kInterfaceCpp

```
const char* const Rcpp::attributes::kInterfaceCpp = "cpp"
```

Definition at line 163 of file attributes.cpp.

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions(), and Rcpp::attributes::ExportsGenerator::writeFunctions().

### 5.5.2.9 kInterfaceR

```
const char* const Rcpp::attributes::kInterfaceR = "r"
```

Definition at line 162 of file attributes.cpp.

Referenced by Rcpp::attributes::REExportsGenerator::doWriteFunctions(), and Rcpp::attributes::SourceFileAttributesParser::hasInterface().

#### 5.5.2.10 kInterfacesAttribute

```
const char* const Rcpp::attributes::kInterfacesAttribute = "interfaces"
```

Definition at line 161 of file attributes.cpp.

Referenced by Rcpp::attributes::SourceFileAttributesParser::hasInterface().

#### 5.5.2.11 kParamBlockEnd

```
const char* const Rcpp::attributes::kParamBlockEnd = "}"
```

Definition at line 169 of file attributes.cpp.

#### 5.5.2.12 kParamBlockStart

```
const char* const Rcpp::attributes::kParamBlockStart = "{"
```

Definition at line 168 of file attributes.cpp.

#### 5.5.2.13 kParamValueFalse

```
const char* const Rcpp::attributes::kParamValueFalse = "false"
```

Definition at line 164 of file attributes.cpp.

#### 5.5.2.14 kParamValueFALSE

```
const char* const Rcpp::attributes::kParamValueFALSE = "FALSE"
```

Definition at line 166 of file attributes.cpp.

### 5.5.2.15 kParamValueTrue

```
const char* const Rcpp::attributes::kParamValueTrue = "true"
```

Definition at line 165 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::invisible(), and Rcpp::attributes::Attribute::rng().

### 5.5.2.16 kParamValueTRUE

```
const char* const Rcpp::attributes::kParamValueTRUE = "TRUE"
```

Definition at line 167 of file attributes.cpp.

Referenced by Rcpp::attributes::Attribute::invisible(), and Rcpp::attributes::Attribute::rng().

### 5.5.2.17 kPluginsAttribute

```
const char* const Rcpp::attributes::kPluginsAttribute = "plugins"
```

Definition at line 160 of file attributes.cpp.

### 5.5.2.18 kWhitespaceChars

```
const char *const Rcpp::attributes::kWhitespaceChars = " \f\n\r\t\v"
```

Definition at line 3022 of file attributes.cpp.

Referenced by isRoxygenCpp(), isWhitespace(), stripTrailingLineComments(), and trimWhitespace().

## 5.6 Rcpp::internal Namespace Reference

internal implementation details

### Namespaces

- [debug](#)



## Classes

- class [InterruptedException](#)
- class [NamedPlaceholder](#)
- struct [UnwindData](#)
- class [string\\_proxy](#)
- class [const\\_string\\_proxy](#)
- class [generic\\_proxy](#)
- class [const\\_generic\\_proxy](#)
- class [simple\\_name\\_proxy](#)
- class [string\\_name\\_proxy](#)
- class [generic\\_name\\_proxy](#)
- class [element\\_converter](#)
- class [string\\_element\\_converter](#)
- class [generic\\_element\\_converter](#)
- class [DimNameProxy](#)
- class [LazyVector](#)
- class [LazyVector< Rcpp::Vector< RTYPE > >](#)
- class [RangelIndexer](#)

## Functions

- `template<> double caster< Rcpp::Date, double > (Rcpp::Date from)`
- `template<> Rcpp::Date caster< double, Rcpp::Date > (double from)`
- `template<> double caster< Rcpp::Datetime, double > (Rcpp::Datetime from)`
- `template<> Rcpp::Datetime caster< double, Rcpp::Datetime > (double from)`
- `void * as\_module\_object\_internal (SEXP obj)`
- `template<typename InputIterator, typename value_type >  
void export\_range\_dispatch (SEXP x, InputIterator first, ::Rcpp::traits::r\_type\_generic\_tag)`
- `bool is\_atomic (SEXP x)`
- `bool is\_matrix (SEXP x)`
- `template<> bool is\_simple< int > (SEXP x)`
- `template<> bool is\_simple< double > (SEXP x)`
- `template<> bool is\_simple< bool > (SEXP x)`
- `template<> bool is\_simple< std::string > (SEXP x)`
- `template<> bool is\_simple< String > (SEXP x)`
- `template<> bool is\_simple< Rcomplex > (SEXP x)`
- `template<> bool is\_simple< CharacterVector > (SEXP x)`
- `template<> bool is\_simple< CharacterMatrix > (SEXP x)`
- `template<> bool is\_simple< RObject > (SEXP)`
- `template<> bool is\_simple< IntegerVector > (SEXP x)`
- `template<> bool is\_simple< ComplexVector > (SEXP x)`
- `template<> bool is\_simple< RawVector > (SEXP x)`
- `template<> bool is\_simple< NumericVector > (SEXP x)`
- `template<> bool is\_simple< LogicalVector > (SEXP x)`
- `template<> bool is\_simple< Language > (SEXP x)`
- `template<> bool is\_simple< DottedPair > (SEXP x)`
- `template<> bool is\_simple< List > (SEXP x)`
- `template<> bool is\_simple< IntegerMatrix > (SEXP x)`
- `template<> bool is\_simple< ComplexMatrix > (SEXP x)`

- `template<> bool is__simple< RawMatrix > (SEXP x)`
- `template<> bool is__simple< NumericMatrix > (SEXP x)`
- `template<> bool is__simple< LogicalMatrix > (SEXP x)`
- `template<> bool is__simple< GenericMatrix > (SEXP x)`
- `template<> bool is__simple< DataFrame > (SEXP x)`
- `template<> bool is__simple< WeakReference > (SEXP x)`
- `template<> bool is__simple< Symbol > (SEXP x)`
- `template<> bool is__simple< S4 > (SEXP x)`
- `template<> bool is__simple< Reference > (SEXP x)`
- `template<> bool is__simple< Promise > (SEXP x)`
- `template<> bool is__simple< Pairlist > (SEXP x)`
- `template<> bool is__simple< Function > (SEXP x)`
- `template<> bool is__simple< Environment > (SEXP x)`
- `template<> bool is__simple< Formula > (SEXP x)`
- `template<> bool is__simple< Date > (SEXP x)`
- `template<> bool is__simple< Datetime > (SEXP x)`
- `template<> bool is__simple< DateVector > (SEXP x)`
- `template<> bool is__simple< DatetimeVector > (SEXP x)`
- `bool is_module_object_internal (SEXP obj, const char *clazz)`
- `template<typename T >`  
`bool is__module__object (SEXP x)`
- `SEXP Rcpp_eval_impl (SEXP expr, SEXP env)`
- `template<typename T >`  
`SEXP wrap_range_sugar_expression (const T &object, Rcpp::traits::true_type)`
- `template<typename InputIterator , typename KEY , typename VALUE , int RTYPE>`  
`SEXP range_wrap_dispatch__impl__pair (InputIterator first, InputIterator last, Rcpp::traits::true_type)`
- `template<typename InputIterator , typename KEY , typename VALUE , int RTYPE>`  
`SEXP range_wrap_dispatch__impl__pair (InputIterator first, InputIterator last, Rcpp::traits::false_type)`
- `template<typename T >`  
`T primitive_as (SEXP x)`
- `template<typename T >`  
`T as (SEXP x, ::Rcpp::traits::r_type_primitive_tag)`
- `const char * check_single_string (SEXP x)`
- `template<typename T >`  
`T as_string (SEXP x, Rcpp::traits::true_type)`
- `template<typename T >`  
`T as_string (SEXP x, Rcpp::traits::false_type)`
- `template<typename T >`  
`T as (SEXP x, ::Rcpp::traits::r_type_string_tag)`
- `template<typename T >`  
`T as (SEXP x, ::Rcpp::traits::r_type_RcppString_tag)`
- `template<typename T >`  
`T as (SEXP x, ::Rcpp::traits::r_type_generic_tag)`
- `template<typename T >`  
`object< T > as_module_object (SEXP x)`
- `template<typename T >`  
`T as (SEXP x, ::Rcpp::traits::r_type_module_object_const_pointer_tag)`
- `template<typename T >`  
`T as (SEXP x, ::Rcpp::traits::r_type_module_object_pointer_tag)`
- `template<typename T >`  
`T as (SEXP x, ::Rcpp::traits::r_type_module_object_tag)`

- `template<typename T >`  
`T as (SEXP x, ::Rcpp::traits::r_type_module_object_reference_tag)`
- `template<typename T >`  
`T as (SEXP x, ::Rcpp::traits::r_type_module_object_const_reference_tag)`
- `template<typename T >`  
`T as (SEXP x, ::Rcpp::traits::r_type_enum_tag)`
- `SEXP empty_data_frame ()`
- `SEXP getPosixClasses ()`
- `SEXP new_posixt_object (double d)`
- `SEXP new_date_object (double d)`
- `SEXP longjumpSentinel (SEXP token)`
- `bool isLongjumpSentinel (SEXP x)`
- `SEXP getLongjumpToken (SEXP sentinel)`
- `void resumeJump (SEXP token)`
- `SEXP nth (SEXP s, int n)`
- `bool is_Rcpp_eval_call (SEXP expr)`
- `template<typename T >`  
`SEXP grow__dispatch (::Rcpp::traits::false_type, const T &head, SEXP tail)`
- `template<typename T >`  
`SEXP grow__dispatch (::Rcpp::traits::true_type, const T &head, SEXP tail)`
- `SEXP interruptedError ()`
- `template<typename T >`  
`bool is__simple (SEXP x)`
- `template<typename T >`  
`bool is__dispatch (SEXP x, Rcpp::traits::false_type)`
- `template<typename T >`  
`bool is__dispatch (SEXP x, Rcpp::traits::true_type)`
- `template<typename FROM , typename TO >`  
`std::string get_converter_name (const char *from, const char *to)`
- `template<typename Class >`  
`SEXP make_new_object (Class *ptr)`
- `SEXP convert_using_rfunction (SEXP x, const char *const fun)`
- `template<int TARGET>`  
`SEXP r_true_cast (SEXP x)`
- `template<int RTYPE>`  
`SEXP basic_cast (SEXP x)`
- `template<> SEXP r_true_cast< INTSXP > (SEXP x)`
- `template<> SEXP r_true_cast< REALSXP > (SEXP x)`
- `template<> SEXP r_true_cast< RAWSXP > (SEXP x)`
- `template<> SEXP r_true_cast< CPLXSXP > (SEXP x)`
- `template<> SEXP r_true_cast< LGLSXP > (SEXP x)`
- `template<> SEXP r_true_cast< STRSXP > (SEXP x)`
- `template<> SEXP r_true_cast< VECSXP > (SEXP x)`
- `template<> SEXP r_true_cast< EXPRSXP > (SEXP x)`
- `template<> SEXP r_true_cast< LISTSXP > (SEXP x)`
- `template<> SEXP r_true_cast< LANGSXP > (SEXP x)`
- `attribute_hidden unsigned long enterRNGScope ()`
- `attribute_hidden unsigned long exitRNGScope ()`
- `attribute_hidden unsigned long beginSuspendRNGSynchronization ()`
- `attribute_hidden unsigned long endSuspendRNGSynchronization ()`
- `attribute_hidden char * get_string_buffer ()`

- attribute\_hidden SEXP [get\\_Rcpp\\_namespace](#) ()
- template<> SEXP [make\\_charsexp](#)< Rcpp::String > (const Rcpp::String &s)
- double [complex\\_\\_Re](#) (Rcomplex x)
- double [complex\\_\\_Im](#) (Rcomplex x)
- double [complex\\_\\_Mod](#) (Rcomplex x)
- Rcomplex [complex\\_\\_Conj](#) (Rcomplex x)
- double [complex\\_\\_Arg](#) (Rcomplex x)
- Rcomplex [complex\\_\\_exp](#) (Rcomplex x)
- Rcomplex [complex\\_\\_log](#) (Rcomplex x)
- Rcomplex [complex\\_\\_sqrt](#) (Rcomplex z)
- Rcomplex [complex\\_\\_cos](#) (Rcomplex z)
- Rcomplex [complex\\_\\_cosh](#) (Rcomplex z)
- Rcomplex [complex\\_\\_sin](#) (Rcomplex z)
- Rcomplex [complex\\_\\_tan](#) (Rcomplex z)
- Rcomplex [complex\\_\\_asin](#) (Rcomplex z)
- Rcomplex [complex\\_\\_acos](#) (Rcomplex z)
- Rcomplex [complex\\_\\_atan](#) (Rcomplex z)
- Rcomplex [complex\\_\\_acosh](#) (Rcomplex z)
- Rcomplex [complex\\_\\_asinh](#) (Rcomplex z)
- Rcomplex [complex\\_\\_atanh](#) (Rcomplex z)
- Rcomplex [complex\\_\\_sinh](#) (Rcomplex z)
- Rcomplex [complex\\_\\_tanh](#) (Rcomplex z)
- double [factorial](#) (double x)
- double [lfactorial](#) (double x)
- template<int RTYPE, bool NA, typename T >  
[Rcpp::Vector](#)< RTYPE > [as\\_vector\\_impl](#) ([MatrixBase](#)< RTYPE, NA, T > &t, [Rcpp::traits::false\\_type](#))
- template<int RTYPE, bool NA, typename T >  
[Rcpp::Vector](#)< RTYPE > [as\\_vector\\_impl](#) ([MatrixBase](#)< RTYPE, NA, T > &t, [Rcpp::traits::true\\_type](#))
- int [get\\_line](#) (int index, int nr)
- int [get\\_column](#) (int index, int nr)
- int [get\\_column](#) (int index, int nr, int i)
- void [maybeJump](#) (void \*unwind\_data, Rboolean jump)
- template<int RT>  
bool [operator](#)< (const [const\\_string\\_proxy](#)< RT > &lhs, const [const\\_string\\_proxy](#)< RT > &rhs)
- template<int RT>  
bool [operator](#)> (const [const\\_string\\_proxy](#)< RT > &lhs, const [const\\_string\\_proxy](#)< RT > &rhs)
- template<int RT>  
bool [operator](#)>= (const [const\\_string\\_proxy](#)< RT > &lhs, const [const\\_string\\_proxy](#)< RT > &rhs)
- template<int RT>  
bool [operator](#)<= (const [const\\_string\\_proxy](#)< RT > &lhs, const [const\\_string\\_proxy](#)< RT > &rhs)
- template<template< class > class StoragePolicy>  
std::ostream & [operator](#)<< (std::ostream &os, const [const\\_string\\_proxy](#)< STRSXP, StoragePolicy > &proxy)
- template<template< class > class StoragePolicy>  
std::string [operator](#)+ (const std::string &x, const [const\\_string\\_proxy](#)< STRSXP, StoragePolicy > &y)
- template<int RT>  
bool [operator](#)< (const [string\\_proxy](#)< RT > &lhs, const [string\\_proxy](#)< RT > &rhs)
- template<int RT>  
bool [operator](#)> (const [string\\_proxy](#)< RT > &lhs, const [string\\_proxy](#)< RT > &rhs)
- template<int RT>  
bool [operator](#)>= (const [string\\_proxy](#)< RT > &lhs, const [string\\_proxy](#)< RT > &rhs)
- template<int RT>  
bool [operator](#)<= (const [string\\_proxy](#)< RT > &lhs, const [string\\_proxy](#)< RT > &rhs)

- `std::ostream & operator<<` (`std::ostream &os`, `const string_proxy< STRSXP > &proxy`)
- `std::string operator+` (`const std::string &x`, `const string_proxy< STRSXP > &y`)
- `template<int RTYPE>`  
SEXP `vector_from_string` (`const std::string &st`)
- `template<int RTYPE>`  
SEXP `vector_from_string_expr` (`const std::string &code`)
- `template<>` SEXP `vector_from_string< EXPRSXP >` (`const std::string &st`)

## Variables

- `int rngSynchronizationSuspended = 0`

### 5.6.1 Detailed Description

internal implementation details

### 5.6.2 Function Documentation

#### 5.6.2.1 `as()` [1/10]

```
template<typename T >
T Rcpp::internal::as (
    SEXP x,
    ::Rcpp::traits::r_type_enum_tag )
```

handling enums by converting to int first

Definition at line 129 of file `as.h`.

#### 5.6.2.2 `as()` [2/10]

```
template<typename T >
T Rcpp::internal::as (
    SEXP x,
    ::Rcpp::traits::r_type_generic_tag )
```

Definition at line 85 of file `as.h`.

References `DEMANGLE`, and `RCPP_DEBUG_1`.

**5.6.2.3 as()** [3/10]

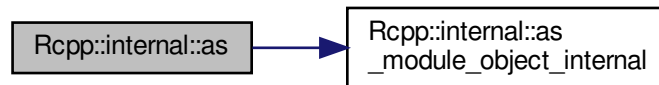
```
template<typename T >
T Rcpp::internal::as (
    SEXP x,
    ::Rcpp::traits::r_type_module_object_const_pointer_tag )
```

handling object<T>

Definition at line 99 of file as.h.

References `as_module_object_internal()`.

Here is the call graph for this function:

**5.6.2.4 as()** [4/10]

```
template<typename T >
T Rcpp::internal::as (
    SEXP x,
    ::Rcpp::traits::r_type_module_object_const_reference_tag )
```

handling T such that T is a reference of a class handled by a module

Definition at line 122 of file as.h.

**5.6.2.5 as()** [5/10]

```
template<typename T >
T Rcpp::internal::as (
    SEXP x,
    ::Rcpp::traits::r_type_module_object_pointer_tag )
```

Definition at line 104 of file as.h.

### 5.6.2.6 as() [6/10]

```
template<typename T >
T Rcpp::internal::as (
    SEXP x,
    ::Rcpp::traits::r_type_module_object_reference_tag )
```

handling T such that T is a reference of a class handled by a module

Definition at line 115 of file as.h.

### 5.6.2.7 as() [7/10]

```
template<typename T >
T Rcpp::internal::as (
    SEXP x,
    ::Rcpp::traits::r_type_module_object_tag )
```

handling T such that T is exposed by a module

Definition at line 109 of file as.h.

### 5.6.2.8 as() [8/10]

```
template<typename T >
T Rcpp::internal::as (
    SEXP x,
    ::Rcpp::traits::r_type_primitive_tag )
```

Definition at line 43 of file as.h.

### 5.6.2.9 as() [9/10]

```
template<typename T >
T Rcpp::internal::as (
    SEXP x,
    ::Rcpp::traits::r_type_RcppString_tag )
```

Definition at line 74 of file as.h.

### 5.6.2.10 as() [10/10]

```
template<typename T >
T Rcpp::internal::as (
    SEXP x,
    ::Rcpp::traits::r_type_string_tag )
```

Definition at line 70 of file as.h.

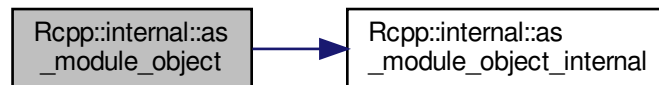
### 5.6.2.11 as\_module\_object()

```
template<typename T >
object<T> Rcpp::internal::as_module_object (
    SEXP x )
```

Definition at line 94 of file as.h.

References `as_module_object_internal()`.

Here is the call graph for this function:



### 5.6.2.12 as\_module\_object\_internal()

```
void * Rcpp::internal::as_module_object_internal (
    SEXP obj ) [inline]
```

Definition at line 24 of file as.h.

Referenced by `as()`, and `as_module_object()`.



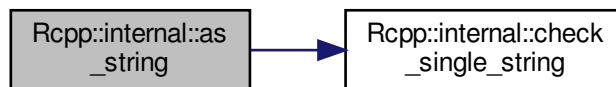
**5.6.2.13 as\_string() [1/2]**

```
template<typename T >
T Rcpp::internal::as_string (
    SEXP x,
    Rcpp::traits::false_type )
```

Definition at line 66 of file as.h.

References check\_single\_string().

Here is the call graph for this function:

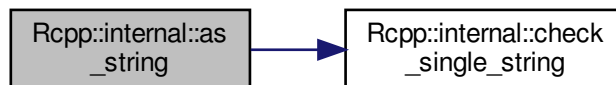
**5.6.2.14 as\_string() [2/2]**

```
template<typename T >
T Rcpp::internal::as_string (
    SEXP x,
    Rcpp::traits::true_type )
```

Definition at line 61 of file as.h.

References check\_single\_string().

Here is the call graph for this function:



### 5.6.2.15 `as_vector__impl()` [1/2]

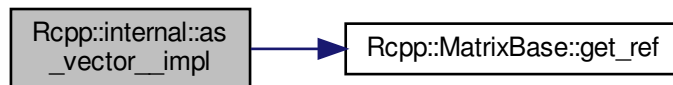
```
template<int RTYPE, bool NA, typename T >
Rcpp::Vector<RTYPE> Rcpp::internal::as_vector__impl (
    MatrixBase< RTYPE, NA, T > & t,
    Rcpp::traits::false_type ) [inline]
```

Definition at line 30 of file `as_vector.h`.

References `Rcpp::MatrixBase< RTYPE, na, MATRIX >::get_ref()`.

Referenced by `Rcpp::as_vector()`.

Here is the call graph for this function:



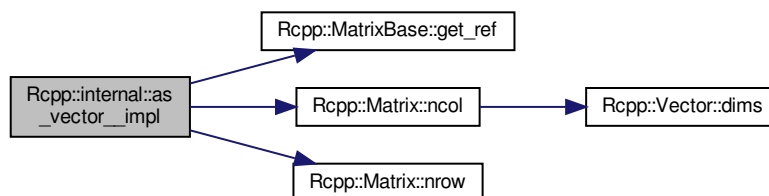
### 5.6.2.16 `as_vector__impl()` [2/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::Vector<RTYPE> Rcpp::internal::as_vector__impl (
    MatrixBase< RTYPE, NA, T > & t,
    Rcpp::traits::true_type ) [inline]
```

Definition at line 44 of file `as_vector.h`.

References `Rcpp::MatrixBase< RTYPE, na, MATRIX >::get_ref()`, `Rcpp::Matrix< RTYPE, StoragePolicy >::ncol()`, and `Rcpp::Matrix< RTYPE, StoragePolicy >::nrow()`.

Here is the call graph for this function:



### 5.6.2.17 basic\_cast()

```
template<int RTYPE>
SEXP Rcpp::internal::basic_cast (
    SEXP x )
```

Definition at line 59 of file r\_cast.h.

### 5.6.2.18 beginSuspendRNGSynchronization()

```
unsigned long Rcpp::internal::beginSuspendRNGSynchronization ( ) [inline]
```

Definition at line 103 of file routines.h.

References GET\_CALLABLE.

Referenced by registerFunctions(), and Rcpp::SuspendRNGSynchronizationScope::SuspendRNGSynchronization←Scope().

### 5.6.2.19 caster< double, Rcpp::Date >()

```
template<>
Rcpp::Date Rcpp::internal::caster< double, Rcpp::Date > (
    double from ) [inline]
```

Definition at line 146 of file Date.h.

### 5.6.2.20 caster< double, Rcpp::Datetime >()

```
template<>
Rcpp::Datetime Rcpp::internal::caster< double, Rcpp::Datetime > (
    double from ) [inline]
```

Definition at line 131 of file Datetime.h.

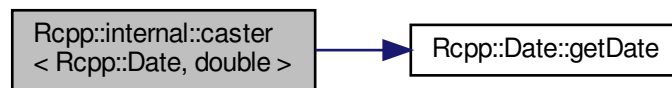
### 5.6.2.21 `caster< Rcpp::Date, double >()`

```
template<>
double Rcpp::internal::caster< Rcpp::Date, double > (
    Rcpp::Date from ) [inline]
```

Definition at line 143 of file Date.h.

References `Rcpp::Date::getDate()`.

Here is the call graph for this function:



### 5.6.2.22 `caster< Rcpp::Datetime, double >()`

```
template<>
double Rcpp::internal::caster< Rcpp::Datetime, double > (
    Rcpp::Datetime from ) [inline]
```

Definition at line 128 of file Datetime.h.

References `Rcpp::Datetime::getFractionalTimestamp()`.

Here is the call graph for this function:



### 5.6.2.23 check\_single\_string()

```
const char* Rcpp::internal::check_single_string (  
    SEXP x ) [inline]
```

Definition at line 47 of file as.h.

Referenced by Rcpp::as< char >(), and as\_string().

### 5.6.2.24 complex\_\_acos()

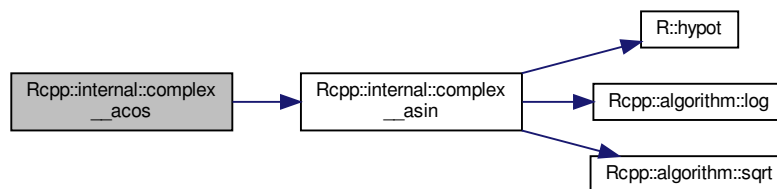
```
Rcomplex Rcpp::internal::complex__acos (  
    Rcomplex z ) [inline]
```

Definition at line 148 of file complex.h.

References complex\_\_asin().

Referenced by complex\_\_acosh().

Here is the call graph for this function:



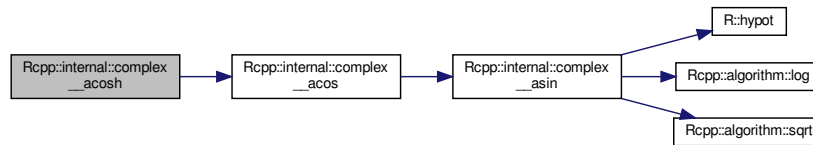
### 5.6.2.25 `complex__acosh()`

```
Rcomplex Rcpp::internal::complex__acosh (
    Rcomplex z ) [inline]
```

Definition at line 181 of file `complex.h`.

References `complex__acos()`.

Here is the call graph for this function:



### 5.6.2.26 `complex__Arg()`

```
double Rcpp::internal::complex__Arg (
    Rcomplex x ) [inline]
```

Definition at line 65 of file `complex.h`.

### 5.6.2.27 `complex__asin()`

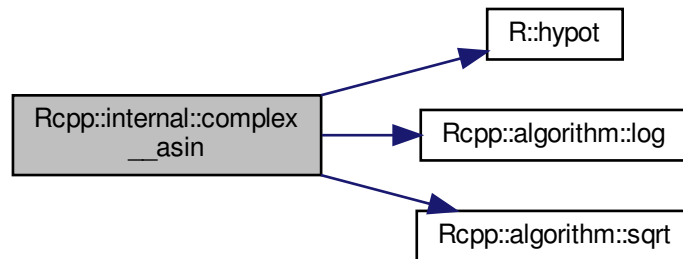
```
Rcomplex Rcpp::internal::complex__asin (
    Rcomplex z ) [inline]
```

Definition at line 132 of file `complex.h`.

References `R::hypot()`, `Rcpp::algorithm::log()`, and `Rcpp::algorithm::sqrt()`.

Referenced by `complex__acos()`, and `complex__asinh()`.

Here is the call graph for this function:



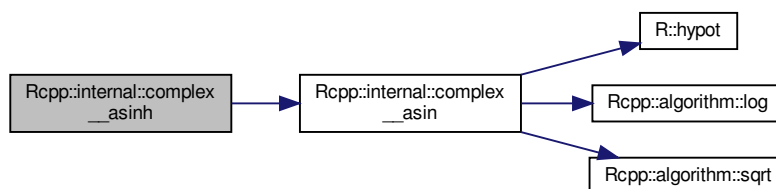
#### 5.6.2.28 `complex__asinh()`

```
Rcomplex Rcpp::internal::complex__asinh (  
    Rcomplex z ) [inline]
```

Definition at line 188 of file `complex.h`.

References `complex__asin()`.

Here is the call graph for this function:



### 5.6.2.29 `complex__atan()`

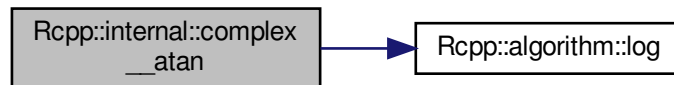
```
Rcomplex Rcpp::internal::complex__atan (  
    Rcomplex z ) [inline]
```

Definition at line 164 of file `complex.h`.

References `Rcpp::algorithm::log()`.

Referenced by `complex__atanh()`.

Here is the call graph for this function:



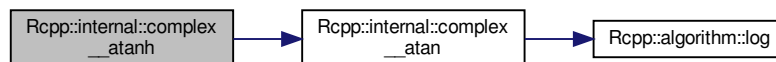
### 5.6.2.30 `complex__atanh()`

```
Rcomplex Rcpp::internal::complex__atanh (  
    Rcomplex z ) [inline]
```

Definition at line 198 of file `complex.h`.

References `complex__atan()`.

Here is the call graph for this function:





### 5.6.2.31 `complex__Conj()`

```
Rcomplex Rcpp::internal::complex__Conj (  
    Rcomplex x ) [inline]
```

Definition at line 59 of file complex.h.

### 5.6.2.32 `complex__cos()`

```
Rcomplex Rcpp::internal::complex__cos (  
    Rcomplex z ) [inline]
```

Definition at line 98 of file complex.h.

### 5.6.2.33 `complex__cosh()`

```
Rcomplex Rcpp::internal::complex__cosh (  
    Rcomplex z ) [inline]
```

Definition at line 104 of file complex.h.

### 5.6.2.34 `complex__exp()`

```
Rcomplex Rcpp::internal::complex__exp (  
    Rcomplex x ) [inline]
```

Definition at line 67 of file complex.h.

References `Rcpp::algorithm::exp()`.

Here is the call graph for this function:



### 5.6.2.35 `complex__Im()`

```
double Rcpp::internal::complex__Im (  
    Rcomplex x ) [inline]
```

Definition at line 57 of file complex.h.

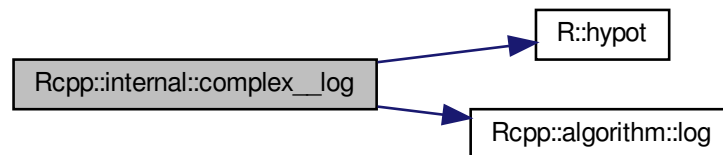
### 5.6.2.36 `complex__log()`

```
Rcomplex Rcpp::internal::complex__log (  
    Rcomplex x ) [inline]
```

Definition at line 74 of file complex.h.

References `R::hypot()`, and `Rcpp::algorithm::log()`.

Here is the call graph for this function:



### 5.6.2.37 `complex__Mod()`

```
double Rcpp::internal::complex__Mod (  
    Rcomplex x ) [inline]
```

Definition at line 58 of file complex.h.

References `Rcpp::algorithm::sqrt()`.

Here is the call graph for this function:



### 5.6.2.38 complex\_\_Re()

```
double Rcpp::internal::complex__Re (  
    Rcomplex x ) [inline]
```

Definition at line 56 of file complex.h.

### 5.6.2.39 complex\_\_sin()

```
Rcomplex Rcpp::internal::complex__sin (  
    Rcomplex z ) [inline]
```

Definition at line 110 of file complex.h.

Referenced by complex\_\_sinh().

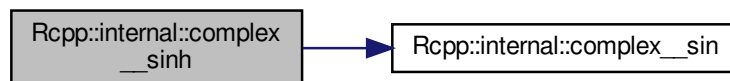
### 5.6.2.40 complex\_\_sinh()

```
Rcomplex Rcpp::internal::complex__sinh (  
    Rcomplex z ) [inline]
```

Definition at line 207 of file complex.h.

References complex\_\_sin().

Here is the call graph for this function:



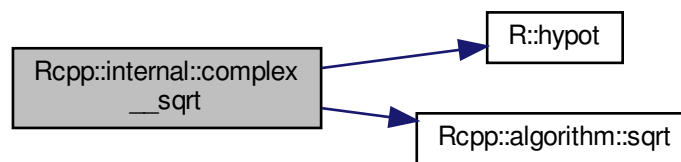
#### 5.6.2.41 `complex__sqrt()`

```
Rcomplex Rcpp::internal::complex__sqrt (  
    Rcomplex z ) [inline]
```

Definition at line 80 of file `complex.h`.

References `R::hypot()`, and `Rcpp::algorithm::sqrt()`.

Here is the call graph for this function:



#### 5.6.2.42 `complex__tan()`

```
Rcomplex Rcpp::internal::complex__tan (  
    Rcomplex z ) [inline]
```

Definition at line 116 of file `complex.h`.

Referenced by `complex__tanh()`.

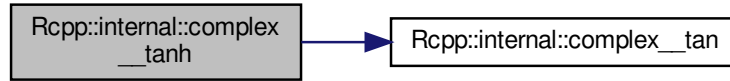
#### 5.6.2.43 `complex__tanh()`

```
Rcomplex Rcpp::internal::complex__tanh (  
    Rcomplex z ) [inline]
```

Definition at line 218 of file `complex.h`.

References `complex__tan()`.

Here is the call graph for this function:



#### 5.6.2.44 convert\_using\_rfunction()

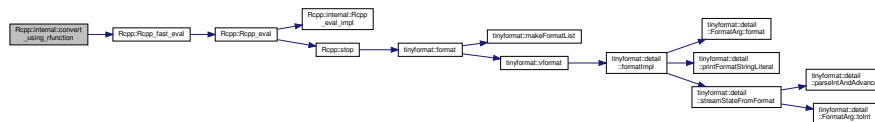
```
SEXP Rcpp::internal::convert_using_rfunction (
    SEXP x,
    const char *const fun ) [inline]
```

Definition at line 30 of file r\_cast.h.

References Rcpp::Rcpp\_fast\_eval().

Referenced by Rcpp::Formula\_Impl(), r\_true\_cast< EXPRXP >(), r\_true\_cast< LANGXP >(), r\_true\_cast< LISTXP >(), r\_true\_cast< VECSXP >(), and Rcpp::DataFrame\_Impl< StoragePolicy >::set\_\_().

Here is the call graph for this function:



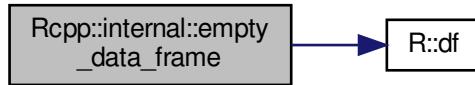
#### 5.6.2.45 empty\_data\_frame()

```
SEXP Rcpp::internal::empty_data_frame ( ) [inline]
```

Definition at line 28 of file DataFrame.h.

References R::df().

Here is the call graph for this function:



#### 5.6.2.46 endSuspendRNGSynchronization()

```
unsigned long Rcpp::internal::endSuspendRNGSynchronization ( ) [inline]
```

Definition at line 109 of file routines.h.

References GET\_CALLABLE.

Referenced by registerFunctions(), and Rcpp::SuspendRNGSynchronizationScope::~~SuspendRNGSynchronizationScope().

#### 5.6.2.47 enterRNGScope()

```
unsigned long Rcpp::internal::enterRNGScope ( ) [inline]
```

Definition at line 91 of file routines.h.

References GET\_CALLABLE.

Referenced by registerFunctions(), and Rcpp::RNGScope::RNGScope().

#### 5.6.2.48 exitRNGScope()

```
unsigned long Rcpp::internal::exitRNGScope ( ) [inline]
```

Definition at line 97 of file routines.h.

References GET\_CALLABLE.

Referenced by registerFunctions(), and Rcpp::RNGScope::~~RNGScope().

### 5.6.2.49 export\_range\_\_dispatch()

```
template<typename InputIterator , typename value_type >
void Rcpp::internal::export_range__dispatch (
    SEXP x,
    InputIterator first,
    ::Rcpp::traits::r_type_generic_tag )
```

Definition at line 29 of file export.h.

### 5.6.2.50 factorial()

```
double Rcpp::internal::factorial (
    double x ) [inline]
```

Definition at line 55 of file math.h.

### 5.6.2.51 get\_column() [1/2]

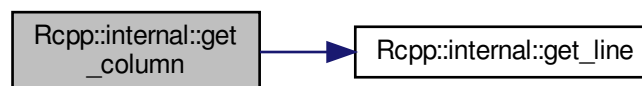
```
int Rcpp::internal::get_column (
    int index,
    int nr ) [inline]
```

Definition at line 32 of file tools.h.

References `get_line()`.

Referenced by `Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update_index()`.

Here is the call graph for this function:



### 5.6.2.52 `get_column()` [2/2]

```
int Rcpp::internal::get_column (
    int index,
    int nr,
    int i ) [inline]
```

Definition at line 37 of file `tools.h`.

### 5.6.2.53 `get_converter_name()`

```
template<typename FROM , typename TO >
std::string Rcpp::internal::get_converter_name (
    const char * from,
    const char * to )
```

Definition at line 32 of file `Module.h`.

### 5.6.2.54 `get_line()`

```
int Rcpp::internal::get_line (
    int index,
    int nr ) [inline]
```

Definition at line 28 of file `tools.h`.

Referenced by `get_column()`, and `Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update_index()`.

### 5.6.2.55 `get_Rcpp_namespace()`

```
SEXP Rcpp::internal::get_Rcpp_namespace ( ) [inline]
```

Definition at line 121 of file `routines.h`.

References `GET_CALLABLE`.

Referenced by `Rcpp::Rcpp_namespace()`, and `registerFunctions()`.



### 5.6.2.56 `get_string_buffer()`

```
char * Rcpp::internal::get_string_buffer ( ) [inline]
```

Definition at line 115 of file `routines.h`.

References `GET_CALLABLE`.

Referenced by `registerFunctions()`.

### 5.6.2.57 `getLongjumpToken()`

```
SEXP Rcpp::internal::getLongjumpToken (
    SEXP sentinel ) [inline]
```

Definition at line 144 of file `exceptions.h`.

Referenced by `Rcpp::LongjumpException::LongjumpException()`, and `resumeJump()`.

### 5.6.2.58 `getPosixClasses()`

```
SEXP Rcpp::internal::getPosixClasses ( ) [inline]
```

Definition at line 179 of file `Date.h`.

Referenced by `new_posixt_object()`, and `Rcpp::wrap_extra_steps< Rcpp::Datetime >()`.

### 5.6.2.59 `grow__dispatch()` [1/2]

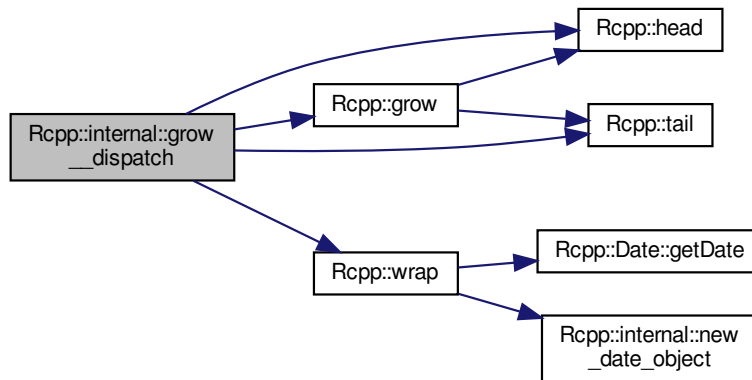
```
template<typename T >
SEXP Rcpp::internal::grow__dispatch (
    ::Rcpp::traits::false_type ,
    const T & head,
    SEXP tail ) [inline]
```

Definition at line 43 of file `grow.h`.

References `Rcpp::grow()`, `Rcpp::head()`, `Rcpp::tail()`, and `Rcpp::wrap()`.

Referenced by `Rcpp::grow()`.

Here is the call graph for this function:



### 5.6.2.60 `grow__dispatch()` [2/2]

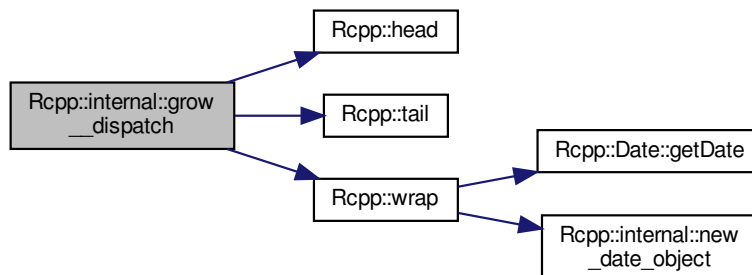
```

template<typename T >
SEXPRcpp::internal::grow__dispatch (
    ::Rcpp::traits::true_type ,
    const T & head,
    SEXPRtail ) [inline]
  
```

Definition at line 48 of file `grow.h`.

References `Rcpp::head()`, `Rcpp::tail()`, and `Rcpp::wrap()`.

Here is the call graph for this function:



### 5.6.2.61 interruptedError()

```
SEXP Rcpp::internal::interruptedError ( ) [inline]
```

Definition at line 40 of file Interrupt.h.

### 5.6.2.62 is\_\_dispatch() [1/2]

```
template<typename T >  
bool Rcpp::internal::is__dispatch (   
    SEXP x,  
    Rcpp::traits::false_type ) [inline]
```

Definition at line 38 of file is.h.

### 5.6.2.63 is\_\_dispatch() [2/2]

```
template<typename T >  
bool Rcpp::internal::is__dispatch (   
    SEXP x,  
    Rcpp::traits::true_type ) [inline]
```

Definition at line 43 of file is.h.

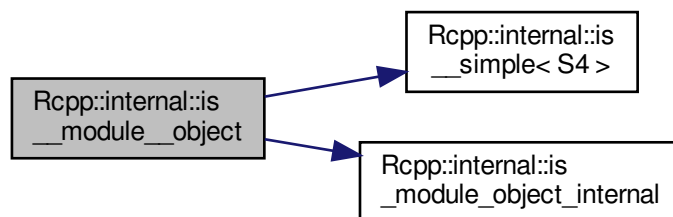
### 5.6.2.64 is\_\_module\_\_object()

```
template<typename T >  
bool Rcpp::internal::is__module__object (   
    SEXP x )
```

Definition at line 160 of file is.h.

References `is__simple< S4 >()`, and `is__module__object__internal()`.

Here is the call graph for this function:



**5.6.2.65 is\_\_simple()**

```
template<typename T >
bool Rcpp::internal::is__simple (
    SEXP x )
```

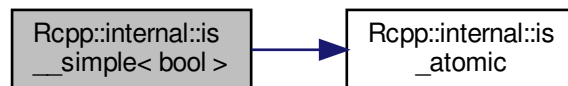
**5.6.2.66 is\_\_simple< bool >()**

```
template<>
bool Rcpp::internal::is__simple< bool > (
    SEXP x ) [inline]
```

Definition at line 39 of file is.h.

References [is\\_atomic\(\)](#).

Here is the call graph for this function:

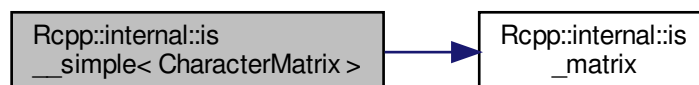
**5.6.2.67 is\_\_simple< CharacterMatrix >()**

```
template<>
bool Rcpp::internal::is__simple< CharacterMatrix > (
    SEXP x ) [inline]
```

Definition at line 54 of file is.h.

References [is\\_matrix\(\)](#).

Here is the call graph for this function:



### 5.6.2.68 is\_\_simple< CharacterVector >()

```
template<>
bool Rcpp::internal::is__simple< CharacterVector > (
    SEXP x ) [inline]
```

Definition at line 51 of file is.h.

### 5.6.2.69 is\_\_simple< ComplexMatrix >()

```
template<>
bool Rcpp::internal::is__simple< ComplexMatrix > (
    SEXP x ) [inline]
```

Definition at line 87 of file is.h.

References is\_matrix().

Here is the call graph for this function:



### 5.6.2.70 is\_\_simple< ComplexVector >()

```
template<>
bool Rcpp::internal::is__simple< ComplexVector > (
    SEXP x ) [inline]
```

Definition at line 63 of file is.h.

### 5.6.2.71 `is__simple< DataFrame >()`

```
template<>
bool Rcpp::internal::is__simple< DataFrame > (
    SEXP x ) [inline]
```

Definition at line 104 of file is.h.

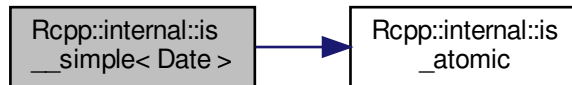
### 5.6.2.72 `is__simple< Date >()`

```
template<>
bool Rcpp::internal::is__simple< Date > (
    SEXP x ) [inline]
```

Definition at line 138 of file is.h.

References `is_atomic()`.

Here is the call graph for this function:



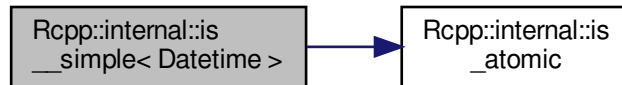
### 5.6.2.73 `is__simple< Datetime >()`

```
template<>
bool Rcpp::internal::is__simple< Datetime > (
    SEXP x ) [inline]
```

Definition at line 141 of file is.h.

References `is_atomic()`.

Here is the call graph for this function:



#### 5.6.2.74 is\_\_simple< DatetimeVector >()

```
template<>
bool Rcpp::internal::is__simple< DatetimeVector > (
    SEXP x ) [inline]
```

Definition at line 147 of file is.h.

#### 5.6.2.75 is\_\_simple< DateVector >()

```
template<>
bool Rcpp::internal::is__simple< DateVector > (
    SEXP x ) [inline]
```

Definition at line 144 of file is.h.

#### 5.6.2.76 is\_\_simple< DottedPair >()

```
template<>
bool Rcpp::internal::is__simple< DottedPair > (
    SEXP x ) [inline]
```

Definition at line 78 of file is.h.

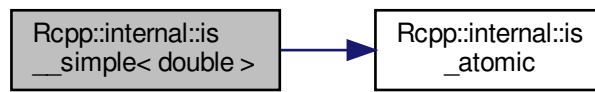
### 5.6.2.77 `is__simple< double >()`

```
template<>
bool Rcpp::internal::is__simple< double > (
    SEXP x ) [inline]
```

Definition at line 36 of file is.h.

References `is_atomic()`.

Here is the call graph for this function:



### 5.6.2.78 `is__simple< Environment >()`

```
template<>
bool Rcpp::internal::is__simple< Environment > (
    SEXP x ) [inline]
```

Definition at line 130 of file is.h.

### 5.6.2.79 `is__simple< Formula >()`

```
template<>
bool Rcpp::internal::is__simple< Formula > (
    SEXP x ) [inline]
```

Definition at line 133 of file is.h.



### 5.6.2.80 is\_\_simple< Function >()

```
template<>
bool Rcpp::internal::is__simple< Function > (
    SEXP x ) [inline]
```

Definition at line 127 of file is.h.

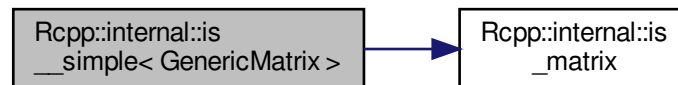
### 5.6.2.81 is\_\_simple< GenericMatrix >()

```
template<>
bool Rcpp::internal::is__simple< GenericMatrix > (
    SEXP x ) [inline]
```

Definition at line 99 of file is.h.

References is\_matrix().

Here is the call graph for this function:



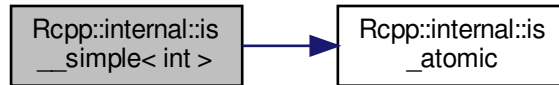
### 5.6.2.82 is\_\_simple< int >()

```
template<>
bool Rcpp::internal::is__simple< int > (
    SEXP x ) [inline]
```

Definition at line 33 of file is.h.

References is\_atomic().

Here is the call graph for this function:



### 5.6.2.83 `is__simple< IntegerMatrix >()`

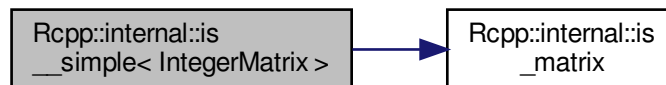
```

template<>
bool Rcpp::internal::is__simple< IntegerMatrix > (
    SEXP x ) [inline]
  
```

Definition at line 84 of file `is.h`.

References `is__matrix()`.

Here is the call graph for this function:



### 5.6.2.84 `is__simple< IntegerVector >()`

```

template<>
bool Rcpp::internal::is__simple< IntegerVector > (
    SEXP x ) [inline]
  
```

Definition at line 60 of file `is.h`.

### 5.6.2.85 is\_\_simple< Language >()

```
template<>
bool Rcpp::internal::is__simple< Language > (
    SEXP x ) [inline]
```

Definition at line 75 of file is.h.

### 5.6.2.86 is\_\_simple< List >()

```
template<>
bool Rcpp::internal::is__simple< List > (
    SEXP x ) [inline]
```

Definition at line 81 of file is.h.

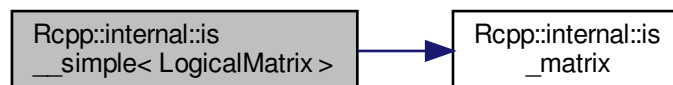
### 5.6.2.87 is\_\_simple< LogicalMatrix >()

```
template<>
bool Rcpp::internal::is__simple< LogicalMatrix > (
    SEXP x ) [inline]
```

Definition at line 96 of file is.h.

References `is_matrix()`.

Here is the call graph for this function:



### 5.6.2.88 `is__simple< LogicalVector >()`

```
template<>
bool Rcpp::internal::is__simple< LogicalVector > (
    SEXP x ) [inline]
```

Definition at line 72 of file is.h.

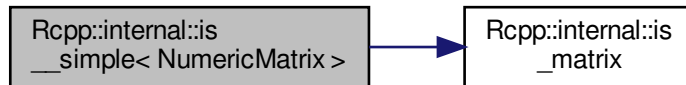
### 5.6.2.89 `is__simple< NumericMatrix >()`

```
template<>
bool Rcpp::internal::is__simple< NumericMatrix > (
    SEXP x ) [inline]
```

Definition at line 93 of file is.h.

References `is_matrix()`.

Here is the call graph for this function:



### 5.6.2.90 `is__simple< NumericVector >()`

```
template<>
bool Rcpp::internal::is__simple< NumericVector > (
    SEXP x ) [inline]
```

Definition at line 69 of file is.h.

### 5.6.2.91 is\_\_simple< Pairlist >()

```
template<>
bool Rcpp::internal::is__simple< Pairlist > (
    SEXP x ) [inline]
```

Definition at line 124 of file is.h.

### 5.6.2.92 is\_\_simple< Promise >()

```
template<>
bool Rcpp::internal::is__simple< Promise > (
    SEXP x ) [inline]
```

Definition at line 121 of file is.h.

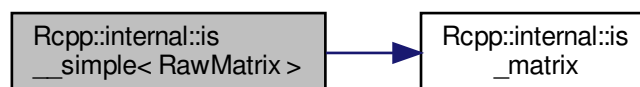
### 5.6.2.93 is\_\_simple< RawMatrix >()

```
template<>
bool Rcpp::internal::is__simple< RawMatrix > (
    SEXP x ) [inline]
```

Definition at line 90 of file is.h.

References [is\\_matrix\(\)](#).

Here is the call graph for this function:



### 5.6.2.94 `is__simple< RawVector >()`

```
template<>
bool Rcpp::internal::is__simple< RawVector > (
    SEXP x ) [inline]
```

Definition at line 66 of file is.h.

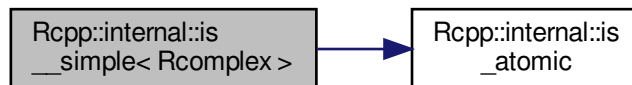
### 5.6.2.95 `is__simple< Rcomplex >()`

```
template<>
bool Rcpp::internal::is__simple< Rcomplex > (
    SEXP x ) [inline]
```

Definition at line 48 of file is.h.

References `is_atomic()`.

Here is the call graph for this function:



### 5.6.2.96 `is__simple< Reference >()`

```
template<>
bool Rcpp::internal::is__simple< Reference > (
    SEXP x ) [inline]
```

Definition at line 117 of file is.h.

### 5.6.2.97 is\_\_simple< RObject >()

```
template<>
bool Rcpp::internal::is__simple< RObject > (
    SEXP ) [inline]
```

Definition at line 57 of file is.h.

### 5.6.2.98 is\_\_simple< S4 >()

```
template<>
bool Rcpp::internal::is__simple< S4 > (
    SEXP x ) [inline]
```

Definition at line 114 of file is.h.

Referenced by is\_\_module\_\_object().

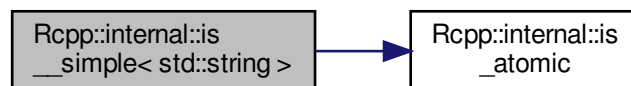
### 5.6.2.99 is\_\_simple< std::string >()

```
template<>
bool Rcpp::internal::is__simple< std::string > (
    SEXP x ) [inline]
```

Definition at line 42 of file is.h.

References is\_atomic().

Here is the call graph for this function:



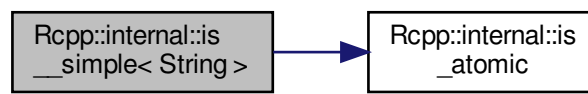
### 5.6.2.100 `is__simple< String >()`

```
template<>
bool Rcpp::internal::is__simple< String > (
    SEXP x ) [inline]
```

Definition at line 45 of file is.h.

References `is_atomic()`.

Here is the call graph for this function:



### 5.6.2.101 `is__simple< Symbol >()`

```
template<>
bool Rcpp::internal::is__simple< Symbol > (
    SEXP x ) [inline]
```

Definition at line 111 of file is.h.

### 5.6.2.102 `is__simple< WeakReference >()`

```
template<>
bool Rcpp::internal::is__simple< WeakReference > (
    SEXP x ) [inline]
```

Definition at line 108 of file is.h.



### 5.6.2.103 is\_atomic()

```
bool Rcpp::internal::is_atomic (
    SEXP x ) [inline]
```

Definition at line 28 of file is.h.

Referenced by `is__simple< bool >()`, `is__simple< Date >()`, `is__simple< Datetime >()`, `is__simple< double >()`, `is__simple< int >()`, `is__simple< Rcomplex >()`, `is__simple< std::string >()`, and `is__simple< String >()`.

### 5.6.2.104 is\_matrix()

```
bool Rcpp::internal::is_matrix (
    SEXP x ) [inline]
```

Definition at line 29 of file is.h.

Referenced by `is__simple< CharacterMatrix >()`, `is__simple< ComplexMatrix >()`, `is__simple< GenericMatrix >()`, `is__simple< IntegerMatrix >()`, `is__simple< LogicalMatrix >()`, `is__simple< NumericMatrix >()`, and `is__simple< RawMatrix >()`.

### 5.6.2.105 is\_module\_object\_internal()

```
bool Rcpp::internal::is_module_object_internal (
    SEXP obj,
    const char * clazz ) [inline]
```

Definition at line 153 of file is.h.

Referenced by `is__module__object()`.

### 5.6.2.106 is\_Rcpp\_eval\_call()

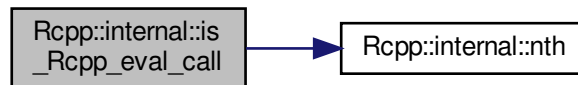
```
bool Rcpp::internal::is_Rcpp_eval_call (
    SEXP expr ) [inline]
```

Definition at line 248 of file exceptions.h.

References nth().

Referenced by get\_last\_call().

Here is the call graph for this function:



### 5.6.2.107 isLongjumpSentinel()

```
bool Rcpp::internal::isLongjumpSentinel (
    SEXP x ) [inline]
```

Definition at line 137 of file exceptions.h.

Referenced by `Rcpp::LongjumpException::LongjumpException()`, and `resumeJump()`.

### 5.6.2.108 lfactorial()

```
double Rcpp::internal::lfactorial (
    double x ) [inline]
```

Definition at line 56 of file math.h.

### 5.6.2.109 longjumpSentinel()

```
SEXP Rcpp::internal::longjumpSentinel (
    SEXP token ) [inline]
```

Definition at line 126 of file exceptions.h.

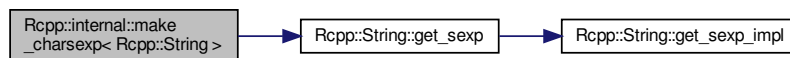
### 5.6.2.110 make\_charsexp< Rcpp::String >()

```
template<>
SEXP Rcpp::internal::make_charsexp< Rcpp::String > (
    const Rcpp::String & s ) [inline]
```

Definition at line 694 of file String.h.

References Rcpp::String::get\_sexp().

Here is the call graph for this function:



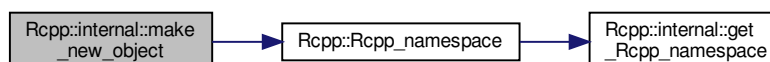
### 5.6.2.111 make\_new\_object()

```
template<typename Class >
SEXP Rcpp::internal::make_new_object (
    Class * ptr )
```

Definition at line 77 of file Module.h.

References Rcpp::Rcpp\_namespace().

Here is the call graph for this function:



### 5.6.2.112 maybeJump()

```
void Rcpp::internal::maybeJump (
    void * unwind_data,
    Rboolean jump ) [inline]
```

Definition at line 41 of file `unwindProtect.h`.

References `Rcpp::internal::UnwindData::jmpbuf`.

Referenced by `Rcpp::unwindProtect()`.

### 5.6.2.113 new\_date\_object()

```
SEXP Rcpp::internal::new_date_object (
    double d ) [inline]
```

Definition at line 192 of file `Date.h`.

Referenced by `Rcpp::wrap()`.

### 5.6.2.114 new\_posixt\_object()

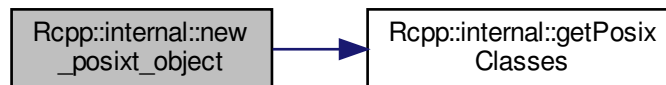
```
SEXP Rcpp::internal::new_posixt_object (
    double d ) [inline]
```

Definition at line 186 of file `Date.h`.

References `getPosixClasses()`.

Referenced by `Rcpp::wrap< Datetime >()`.

Here is the call graph for this function:



**5.6.2.115 nth()**

```
SEXP Rcpp::internal::nth (
    SEXP s,
    int n ) [inline]
```

Definition at line 241 of file exceptions.h.

Referenced by `is_Rcpp_eval_call()`.

**5.6.2.116 operator+() [1/2]**

```
template<template< class > class StoragePolicy>
std::string Rcpp::internal::operator+ (
    const std::string & x,
    const const_string_proxy< STRSXP, StoragePolicy > & y ) [inline]
```

Definition at line 168 of file const\_string\_proxy.h.

**5.6.2.117 operator+() [2/2]**

```
std::string Rcpp::internal::operator+ (
    const std::string & x,
    const string_proxy< STRSXP > & y ) [inline]
```

Definition at line 259 of file string\_proxy.h.

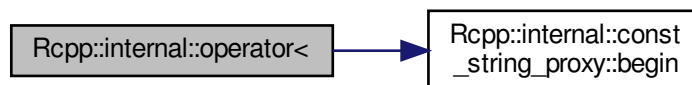
**5.6.2.118 operator<() [1/2]**

```
template<int RT>
bool Rcpp::internal::operator< (
    const const_string_proxy< RT > & lhs,
    const const_string_proxy< RT > & rhs )
```

Definition at line 128 of file const\_string\_proxy.h.

References `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin()`.

Here is the call graph for this function:



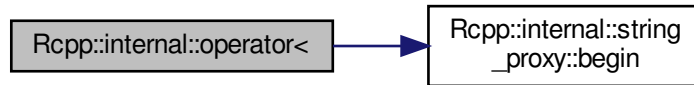
**5.6.2.119 operator<>() [2/2]**

```
template<int RT>
bool Rcpp::internal::operator< (
    const string_proxy< RT > & lhs,
    const string_proxy< RT > & rhs )
```

Definition at line 221 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:

**5.6.2.120 operator<<>() [1/2]**

```
template<template< class > class StoragePolicy>
std::ostream& Rcpp::internal::operator<< (
    std::ostream & os,
    const const_string_proxy< STRSXP, StoragePolicy > & proxy ) [inline]
```

Definition at line 162 of file const\_string\_proxy.h.

**5.6.2.121 operator<<>() [2/2]**

```
std::ostream& Rcpp::internal::operator<< (
    std::ostream & os,
    const string_proxy< STRSXP > & proxy ) [inline]
```

Definition at line 254 of file string\_proxy.h.

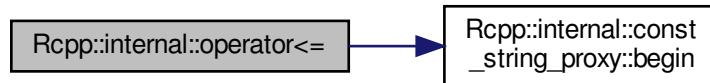
**5.6.2.122 operator<=()** [1/2]

```
template<int RT>
bool Rcpp::internal::operator<= (
    const const_string_proxy< RT > & lhs,
    const const_string_proxy< RT > & rhs )
```

Definition at line 152 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:

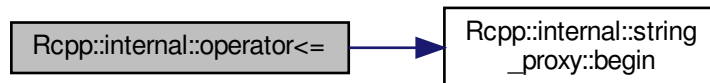
**5.6.2.123 operator<=()** [2/2]

```
template<int RT>
bool Rcpp::internal::operator<= (
    const string_proxy< RT > & lhs,
    const string_proxy< RT > & rhs )
```

Definition at line 245 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



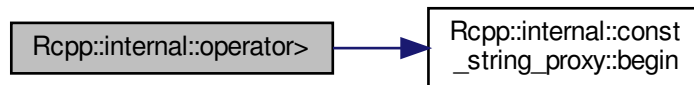
### 5.6.2.124 operator>() [1/2]

```
template<int RT>
bool Rcpp::internal::operator> (
    const const_string_proxy< RT > & lhs,
    const const_string_proxy< RT > & rhs )
```

Definition at line 136 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



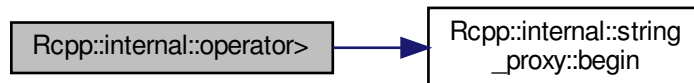
### 5.6.2.125 operator>() [2/2]

```
template<int RT>
bool Rcpp::internal::operator> (
    const string_proxy< RT > & lhs,
    const string_proxy< RT > & rhs )
```

Definition at line 229 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:





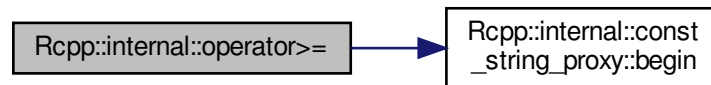
**5.6.2.126 operator>=()** [1/2]

```
template<int RT>
bool Rcpp::internal::operator>= (
    const const_string_proxy< RT > & lhs,
    const const_string_proxy< RT > & rhs )
```

Definition at line 144 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:

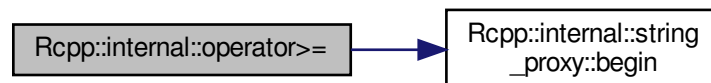
**5.6.2.127 operator>=()** [2/2]

```
template<int RT>
bool Rcpp::internal::operator>= (
    const string_proxy< RT > & lhs,
    const string_proxy< RT > & rhs )
```

Definition at line 237 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



**5.6.2.128 primitive\_as()**

```
template<typename T >
T Rcpp::internal::primitive_as (
    SEXP x )
```

Definition at line 31 of file as.h.

**5.6.2.129 r\_true\_cast()**

```
template<int TARGET>
SEXP Rcpp::internal::r_true_cast (
    SEXP x )
```

Definition at line 46 of file r\_cast.h.

**5.6.2.130 r\_true\_cast< CPLXSXP >()**

```
template<>
SEXP Rcpp::internal::r_true_cast< CPLXSXP > (
    SEXP x ) [inline]
```

Definition at line 98 of file r\_cast.h.

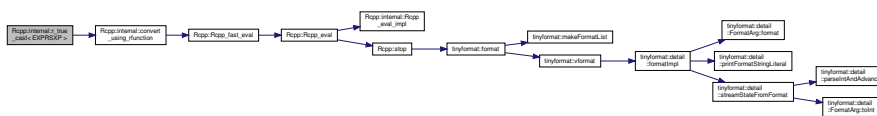
**5.6.2.131 r\_true\_cast< EXPRSXP >()**

```
template<>
SEXP Rcpp::internal::r_true_cast< EXPRSXP > (
    SEXP x ) [inline]
```

Definition at line 141 of file r\_cast.h.

References `convert_using_rfunction()`.

Here is the call graph for this function:



### 5.6.2.132 `r_true_cast<INTSXP>()`

```
template<>
SEXP Rcpp::internal::r_true_cast< INTSXP > (
    SEXP x ) [inline]
```

Definition at line 86 of file `r_cast.h`.

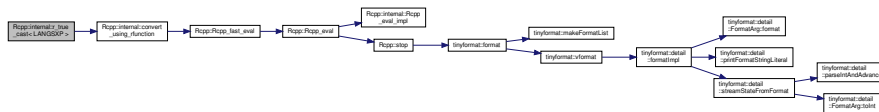
### 5.6.2.133 `r_true_cast<LANGSXP>()`

```
template<>
SEXP Rcpp::internal::r_true_cast< LANGSXP > (
    SEXP x ) [inline]
```

Definition at line 158 of file `r_cast.h`.

References `convert_using_rfunction()`.

Here is the call graph for this function:



### 5.6.2.134 `r_true_cast<LGLSXP>()`

```
template<>
SEXP Rcpp::internal::r_true_cast< LGLSXP > (
    SEXP x ) [inline]
```

Definition at line 102 of file `r_cast.h`.

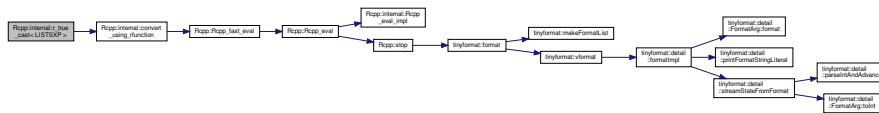
### 5.6.2.135 `r_true_cast<LISTSXP>()`

```
template<>
SEXP Rcpp::internal::r_true_cast<LISTSXP> (
    SEXP x ) [inline]
```

Definition at line 145 of file `r_cast.h`.

References `convert_using_rfunction()`.

Here is the call graph for this function:



### 5.6.2.136 `r_true_cast<RAWSXP>()`

```
template<>
SEXP Rcpp::internal::r_true_cast<RAWSXP> (
    SEXP x ) [inline]
```

Definition at line 94 of file `r_cast.h`.

### 5.6.2.137 `r_true_cast<REALSXP>()`

```
template<>
SEXP Rcpp::internal::r_true_cast<REALSXP> (
    SEXP x ) [inline]
```

Definition at line 90 of file `r_cast.h`.

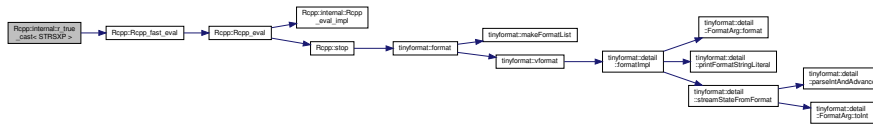
## 5.6.2.138 r\_true\_cast&lt; STRSXP &gt;()

```
template<>
SEXP Rcpp::internal::r_true_cast< STRSXP > (
    SEXP x ) [inline]
```

Definition at line 107 of file r\_cast.h.

References Rcpp::Rcpp\_fast\_eval().

Here is the call graph for this function:



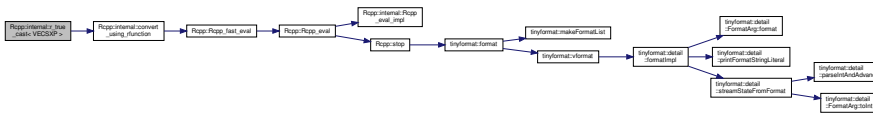
## 5.6.2.139 r\_true\_cast&lt; VECSXP &gt;()

```
template<>
SEXP Rcpp::internal::r_true_cast< VECSXP > (
    SEXP x ) [inline]
```

Definition at line 137 of file r\_cast.h.

References convert\_using\_rfunction().

Here is the call graph for this function:



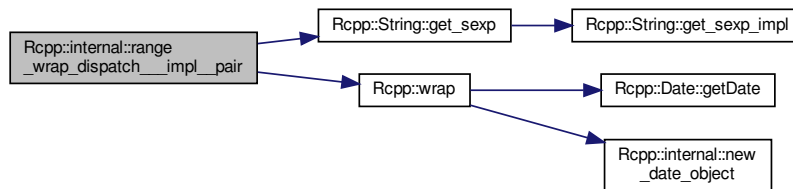
### 5.6.2.140 range\_wrap\_dispatch\_\_\_impl\_\_pair() [1/2]

```
template<typename InputIterator , typename KEY , typename VALUE , int RTYPE>
SEXP Rcpp::internal::range_wrap_dispatch___impl__pair (
    InputIterator first,
    InputIterator last,
    Rcpp::traits::false_type ) [inline]
```

Definition at line 47 of file wrap.h.

References Rcpp::String::get\_sexp(), and Rcpp::wrap().

Here is the call graph for this function:



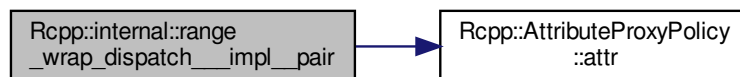
### 5.6.2.141 range\_wrap\_dispatch\_\_\_impl\_\_pair() [2/2]

```
template<typename InputIterator , typename KEY , typename VALUE , int RTYPE>
SEXP Rcpp::internal::range_wrap_dispatch___impl__pair (
    InputIterator first,
    InputIterator last,
    Rcpp::traits::true_type ) [inline]
```

Definition at line 29 of file wrap.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), DEMANGLE, and RCPP\_DEBUG\_3.

Here is the call graph for this function:



### 5.6.2.142 Rcpp\_eval\_impl()

```
SEXP Rcpp::internal::Rcpp_eval_impl (
    SEXP expr,
    SEXP env ) [inline]
```

Definition at line 48 of file Rcpp\_eval.h.

Referenced by Rcpp::fast\_eval(), Rcpp::find(), Rcpp::get(), and Rcpp::Rcpp\_eval().

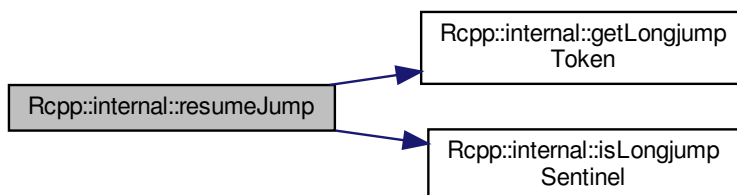
### 5.6.2.143 resumeJump()

```
void Rcpp::internal::resumeJump (
    SEXP token ) [inline]
```

Definition at line 148 of file exceptions.h.

References getLongjumpToken(), and isLongjumpSentinel().

Here is the call graph for this function:



### 5.6.2.144 vector\_from\_string()

```
template<int RTYPE>
SEXP Rcpp::internal::vector_from_string (
    const std::string & st ) [inline]
```

Definition at line 29 of file vector\_from\_string.h.

**5.6.2.145 vector\_from\_string<EXPRXP>()**

```
template<>
SEXP Rcpp::internal::vector_from_string< EXPRXP > (
    const std::string & st ) [inline]
```

Definition at line 50 of file vector\_from\_string.h.

**5.6.2.146 vector\_from\_string\_expr()**

```
template<int RTYPE>
SEXP Rcpp::internal::vector_from_string_expr (
    const std::string & code )
```

Definition at line 35 of file vector\_from\_string.h.

**5.6.2.147 wrap\_range\_sugar\_expression()**

```
template<typename T >
SEXP Rcpp::internal::wrap_range_sugar_expression (
    const T & object,
    Rcpp::traits::true_type ) [inline]
```

Definition at line 30 of file Vector.h.

References DEMANGLE, and RCPP\_DEBUG\_1.

**5.6.3 Variable Documentation****5.6.3.1 rngSynchronizationSuspended**

```
int Rcpp::internal::rngSynchronizationSuspended = 0
```

Definition at line 53 of file api.cpp.

**5.7 Rcpp::internal::debug Namespace Reference****Functions**

- std::string [short\\_file\\_name](#) (const char \*file)



## 5.7.1 Function Documentation

### 5.7.1.1 short\_file\_name()

```
std::string Rcpp::internal::debug::short_file_name (  
    const char * file ) [inline]
```

Definition at line 31 of file macros.h.

Referenced by registerFunctions().

## 5.8 Rcpp::InternalFunctionWithStdFunction Namespace Reference

### Classes

- class [CppFunctionBaseFromStdFunction](#)
- class [CppFunctionBaseFromStdFunction< void, Args... >](#)

## 5.9 Rcpp::stats Namespace Reference

### Classes

- class [D0](#)
- class [D1](#)
- class [D2](#)
- class [D3](#)
- class [P0](#)
- class [P1](#)
- class [P2](#)
- class [P3](#)
- class [Q0](#)
- class [Q1](#)
- class [Q2](#)
- class [Q3](#)
- class [BetaGenerator](#)
- class [BinomGenerator](#)
- class [CauchyGenerator](#)
- class [CauchyGenerator\\_1](#)
- class [CauchyGenerator\\_0](#)
- class [ChisqGenerator](#)
- class [ExpGenerator](#)
- class [ExpGenerator\\_\\_rate1](#)
- class [FGenerator\\_Finite\\_Finite](#)

- class [FGenerator\\_NotFinite\\_Finite](#)
- class [FGenerator\\_Finite\\_NotFinite](#)
- class [GammaGenerator](#)
- class [GeomGenerator](#)
- class [HyperGenerator](#)
- class [LNormGenerator](#)
- class [LNormGenerator\\_1](#)
- class [LNormGenerator\\_0](#)
- class [LogisGenerator](#)
- class [LogisGenerator\\_1](#)
- class [LogisGenerator\\_0](#)
- class [NBinomGenerator](#)
- class [NBinomGenerator\\_Mu](#)
- class [NChisqGenerator](#)
- class [NormGenerator](#)
- class [NormGenerator\\_\\_sd1](#)
- class [NormGenerator\\_\\_mean0](#)
- class [NormGenerator\\_\\_mean0\\_\\_sd1](#)
- class [PoissonGenerator](#)
- class [SignRankGenerator](#)
- class [TGenerator](#)
- class [UnifGenerator](#)
- class [UnifGenerator\\_0\\_\\_1](#)
- class [WeibullGenerator](#)
- class [WeibullGenerator\\_\\_scale1](#)
- class [WilcoxGenerator](#)

## Functions

- double [dcauchy\\_0](#) (double x, int [give\\_log](#))
- double [dcauchy\\_1](#) (double x, double location, int [give\\_log](#))
- double [pcauchy\\_0](#) (double x, int lower\_tail, int log\_p)
- double [pcauchy\\_1](#) (double x, double location, int lower\_tail, int log\_p)
- double [qcauchy\\_0](#) (double p, int lower\_tail, int log\_p)
- double [qcauchy\\_1](#) (double p, double location, int lower\_tail, int log\_p)
- double [d\\_exp\\_0](#) (double x, int [give\\_log](#))
- double [q\\_exp\\_0](#) (double p, int lower\_tail, int log\_p)
- double [p\\_exp\\_0](#) (double x, int lower\_tail, int log\_p)
- double [dgamma\\_1](#) (double x, double shape, int log\_p)
- double [pgamma\\_1](#) (double x, double alpha, int lower\_tail, int log\_p)
- double [qgamma\\_1](#) (double p, double alpha, int lower\_tail, int log\_p)
- double [dlnorm\\_0](#) (double x, int log\_p)
- double [dlnorm\\_1](#) (double x, double meanlog, int log\_p)
- double [plnorm\\_0](#) (double x, int lower\_tail, int log\_p)
- double [plnorm\\_1](#) (double x, double meanlog, int lower\_tail, int log\_p)
- double [qlnorm\\_0](#) (double p, int lower\_tail, int log\_p)
- double [qlnorm\\_1](#) (double p, double meanlog, int lower\_tail, int log\_p)
- double [dlogis\\_0](#) (double x, int [give\\_log](#))
- double [dlogis\\_1](#) (double x, double location, int [give\\_log](#))
- double [plogis\\_0](#) (double x, int lower\_tail, int log\_p)

- double [plogis\\_1](#) (double x, double location, int lower\_tail, int log\_p)
- double [qlogis\\_0](#) (double p, int lower\_tail, int log\_p)
- double [qlogis\\_1](#) (double p, double location, int lower\_tail, int log\_p)
- double [dnorm\\_1](#) (double x, double mu, int [give\\_log](#))
- double [dnorm\\_0](#) (double x, int [give\\_log](#))
- double [pnorm\\_1](#) (double x, double mu, int lower\_tail, int log\_p)
- double [pnorm\\_0](#) (double x, int lower\_tail, int log\_p)
- double [qnorm\\_1](#) (double p, double mu, int lower\_tail, int log\_p)
- double [qnorm\\_0](#) (double p, int lower\_tail, int log\_p)
- double [dunif\\_1](#) (double x, double a, int [give\\_log](#))
- double [dunif\\_0](#) (double x, int [give\\_log](#))
- double [punif\\_1](#) (double x, double a, int lower\_tail, int log\_p)
- double [punif\\_0](#) (double x, int lower\_tail, int log\_p)
- double [qunif\\_1](#) (double p, double a, int lower\_tail, int log\_p)
- double [qunif\\_0](#) (double p, int lower\_tail, int log\_p)
- double [dweibull\\_1](#) (double x, double shape, int [give\\_log](#))
- double [pweibull\\_1](#) (double x, double shape, int lower\_tail, int log\_p)
- double [qweibull\\_1](#) (double p, double shape, int lower\_tail, int log\_p)

## 5.9.1 Function Documentation

### 5.9.1.1 d\_exp\_0()

```
double Rcpp::stats::d_exp_0 (  
    double x,  
    int give_log ) [inline]
```

Definition at line 30 of file exp.h.

References [Rcpp::algorithm::exp\(\)](#), [give\\_log](#), and [R\\_D\\_\\_0](#).

Here is the call graph for this function:



### 5.9.1.2 dcauchy\_0()

```
double Rcpp::stats::dcauchy_0 (  
    double x,  
    int give_log ) [inline]
```

Definition at line 28 of file `cauchy.h`.

References `give_log`.

### 5.9.1.3 dcauchy\_1()

```
double Rcpp::stats::dcauchy_1 (  
    double x,  
    double location,  
    int give_log ) [inline]
```

Definition at line 32 of file `cauchy.h`.

References `give_log`.

### 5.9.1.4 dgamma\_1()

```
double Rcpp::stats::dgamma_1 (  
    double x,  
    double shape,  
    int log_p ) [inline]
```

Definition at line 30 of file `gamma.h`.

References `Rcpp::algorithm::log()`, `ML_POSINF`, and `R_D__0`.

Here is the call graph for this function:



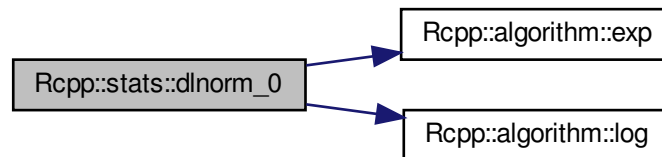
### 5.9.1.5 dlnorm\_0()

```
double Rcpp::stats::dlnorm_0 (  
    double x,  
    int log_p ) [inline]
```

Definition at line 30 of file `lnorm.h`.

References `Rcpp::algorithm::exp()`, `Rcpp::algorithm::log()`, and `R_D__0`.

Here is the call graph for this function:



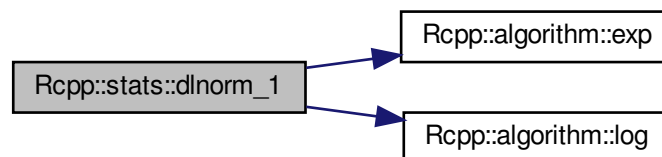
### 5.9.1.6 dlnorm\_1()

```
double Rcpp::stats::dlnorm_1 (  
    double x,  
    double meanlog,  
    int log_p ) [inline]
```

Definition at line 47 of file `lnorm.h`.

References `Rcpp::algorithm::exp()`, `Rcpp::algorithm::log()`, and `R_D__0`.

Here is the call graph for this function:



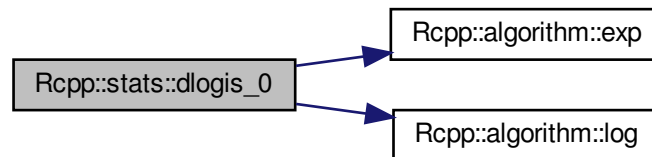
### 5.9.1.7 `dlogis_0()`

```
double Rcpp::stats::dlogis_0 (  
    double x,  
    int give_log ) [inline]
```

Definition at line 28 of file `logis.h`.

References `Rcpp::algorithm::exp()`, `give_log`, and `Rcpp::algorithm::log()`.

Here is the call graph for this function:



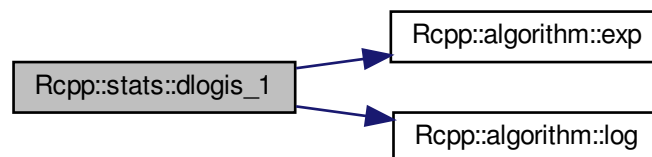
### 5.9.1.8 `dlogis_1()`

```
double Rcpp::stats::dlogis_1 (  
    double x,  
    double location,  
    int give_log ) [inline]
```

Definition at line 40 of file `logis.h`.

References `Rcpp::algorithm::exp()`, `give_log`, and `Rcpp::algorithm::log()`.

Here is the call graph for this function:



### 5.9.1.9 dnorm\_0()

```
double Rcpp::stats::dnorm_0 (  
    double x,  
    int give_log ) [inline]
```

Definition at line 43 of file norm.h.

References Rcpp::algorithm::exp(), give\_log, and R\_D\_\_0.

Here is the call graph for this function:



### 5.9.1.10 dnorm\_1()

```
double Rcpp::stats::dnorm_1 (  
    double x,  
    double mu,  
    int give_log ) [inline]
```

Definition at line 28 of file norm.h.

References Rcpp::algorithm::exp(), give\_log, ML\_NAN, and R\_D\_\_0.

Here is the call graph for this function:



**5.9.1.11 dunif\_0()**

```
double Rcpp::stats::dunif_0 (
    double x,
    int give_log ) [inline]
```

Definition at line 34 of file unif.h.

References `give_log`, and `R_D__0`.

**5.9.1.12 dunif\_1()**

```
double Rcpp::stats::dunif_1 (
    double x,
    double a,
    int give_log ) [inline]
```

Definition at line 30 of file unif.h.

References `give_log`.

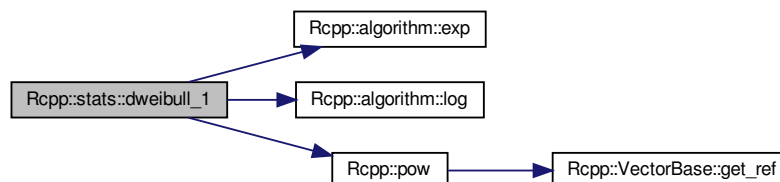
**5.9.1.13 dweibull\_1()**

```
double Rcpp::stats::dweibull_1 (
    double x,
    double shape,
    int give_log ) [inline]
```

Definition at line 30 of file weibull.h.

References `Rcpp::algorithm::exp()`, `give_log`, `Rcpp::algorithm::log()`, `ML_POSINF`, `Rcpp::pow()`, and `R_D__0`.

Here is the call graph for this function:





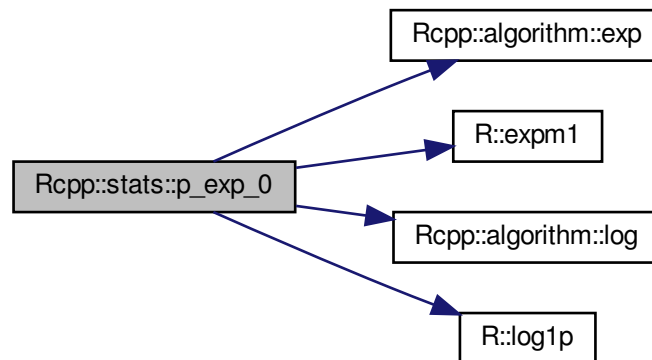
### 5.9.1.14 p\_exp\_0()

```
double Rcpp::stats::p_exp_0 (  
    double x,  
    int lower_tail,  
    int log_p ) [inline]
```

Definition at line 54 of file exp.h.

References [Rcpp::algorithm::exp\(\)](#), [R::expm1\(\)](#), [Rcpp::algorithm::log\(\)](#), [R::log1p\(\)](#), [R\\_D\\_exp](#), and [R\\_DT\\_0](#).

Here is the call graph for this function:



### 5.9.1.15 pcauchy\_0()

```
double Rcpp::stats::pcauchy_0 (  
    double x,  
    int lower_tail,  
    int log_p ) [inline]
```

Definition at line 36 of file cauchy.h.

### 5.9.1.16 pcauchy\_1()

```
double Rcpp::stats::pcauchy_1 (
    double x,
    double location,
    int lower_tail,
    int log_p ) [inline]
```

Definition at line 40 of file `cauchy.h`.

### 5.9.1.17 pgamma\_1()

```
double Rcpp::stats::pgamma_1 (
    double x,
    double alph,
    int lower_tail,
    int log_p ) [inline]
```

Definition at line 56 of file `gamma.h`.

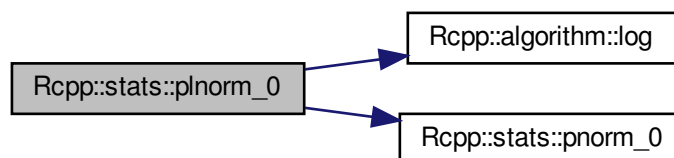
### 5.9.1.18 plnorm\_0()

```
double Rcpp::stats::plnorm_0 (
    double x,
    int lower_tail,
    int log_p ) [inline]
```

Definition at line 65 of file `lnorm.h`.

References `Rcpp::algorithm::log()`, `plnorm_0()`, and `R_DT_0`.

Here is the call graph for this function:



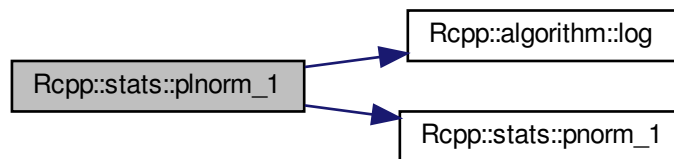
### 5.9.1.19 plnorm\_1()

```
double Rcpp::stats::plnorm_1 (
    double x,
    double meanlog,
    int lower_tail,
    int log_p ) [inline]
```

Definition at line 76 of file `lnorm.h`.

References `Rcpp::algorithm::log()`, `pnorm_1()`, and `R_DT_0`.

Here is the call graph for this function:



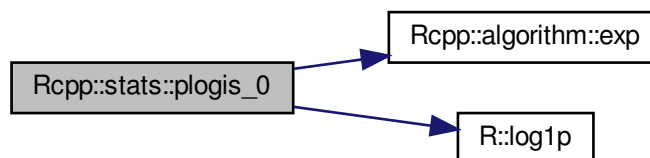
### 5.9.1.20 plogis\_0()

```
double Rcpp::stats::plogis_0 (
    double x,
    int lower_tail,
    int log_p ) [inline]
```

Definition at line 54 of file `logis.h`.

References `Rcpp::algorithm::exp()`, `R::log1p()`, and `R_P_bounds_Inf_01`.

Here is the call graph for this function:



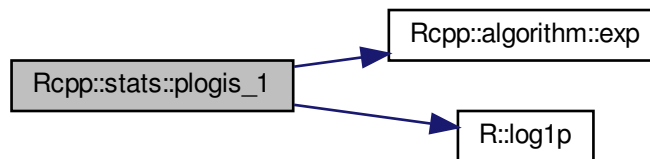
### 5.9.1.21 plogis\_1()

```
double Rcpp::stats::plogis_1 (
    double x,
    double location,
    int lower_tail,
    int log_p ) [inline]
```

Definition at line 69 of file logis.h.

References `Rcpp::algorithm::exp()`, `R::log1p()`, and `R_P_bounds_Inf_01`.

Here is the call graph for this function:



### 5.9.1.22 pnorm\_0()

```
double Rcpp::stats::pnorm_0 (
    double x,
    int lower_tail,
    int log_p ) [inline]
```

Definition at line 78 of file norm.h.

References `R_DT_0`, and `R_DT_1`.

Referenced by `pnorm_0()`.

### 5.9.1.23 pnorm\_1()

```
double Rcpp::stats::pnorm_1 (
    double x,
    double mu,
    int lower_tail,
    int log_p ) [inline]
```

Definition at line 56 of file norm.h.

References ML\_NAN, R\_DT\_0, and R\_DT\_1.

Referenced by plnorm\_1().

### 5.9.1.24 punif\_0()

```
double Rcpp::stats::punif_0 (
    double x,
    int lower_tail,
    int log_p ) [inline]
```

Definition at line 50 of file unif.h.

References R\_D\_val, R\_DT\_0, and R\_DT\_1.

### 5.9.1.25 punif\_1()

```
double Rcpp::stats::punif_1 (
    double x,
    double a,
    int lower_tail,
    int log_p ) [inline]
```

Definition at line 46 of file unif.h.

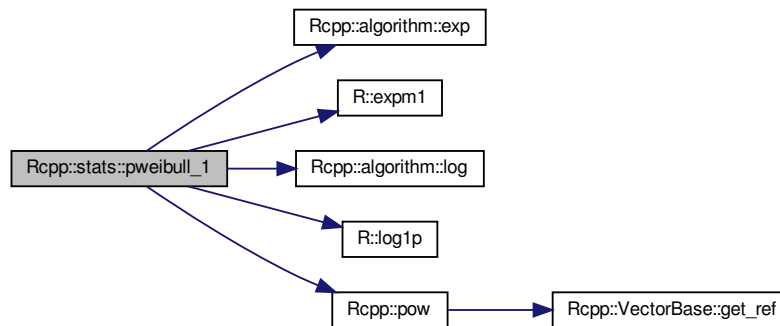
### 5.9.1.26 pweibull\_1()

```
double Rcpp::stats::pweibull_1 (
    double x,
    double shape,
    int lower_tail,
    int log_p ) [inline]
```

Definition at line 49 of file weibull.h.

References `Rcpp::algorithm::exp()`, `R::expm1()`, `Rcpp::algorithm::log()`, `R::log1p()`, `Rcpp::pow()`, `R_D_exp`, and `R_DT_0`.

Here is the call graph for this function:



### 5.9.1.27 q\_exp\_0()

```
double Rcpp::stats::q_exp_0 (
    double p,
    int lower_tail,
    int log_p ) [inline]
```

Definition at line 42 of file exp.h.

References `R_DT_0`, and `R_DT_Clog`.

### 5.9.1.28 qcauchy\_0()

```
double Rcpp::stats::qcauchy_0 (  
    double p,  
    int lower_tail,  
    int log_p ) [inline]
```

Definition at line 44 of file `cauchy.h`.

### 5.9.1.29 qcauchy\_1()

```
double Rcpp::stats::qcauchy_1 (  
    double p,  
    double location,  
    int lower_tail,  
    int log_p ) [inline]
```

Definition at line 47 of file `cauchy.h`.

### 5.9.1.30 qgamma\_1()

```
double Rcpp::stats::qgamma_1 (  
    double p,  
    double alpha,  
    int lower_tail,  
    int log_p ) [inline]
```

Definition at line 59 of file `gamma.h`.

### 5.9.1.31 qlnorm\_0()

```
double Rcpp::stats::qlnorm_0 (  
    double p,  
    int lower_tail,  
    int log_p ) [inline]
```

Definition at line 87 of file `lnorm.h`.

References `Rcpp::algorithm::exp()`, `ML_POSINF`, and `R_Q_P01_boundaries`.

Here is the call graph for this function:



### 5.9.1.32 `qlnorm_1()`

```
double Rcpp::stats::qlnorm_1 (  
    double p,  
    double meanlog,  
    int lower_tail,  
    int log_p ) [inline]
```

Definition at line 97 of file `lnorm.h`.

References `Rcpp::algorithm::exp()`, `ML_POSINF`, and `R_Q_P01_boundaries`.

Here is the call graph for this function:



### 5.9.1.33 `qlogis_0()`

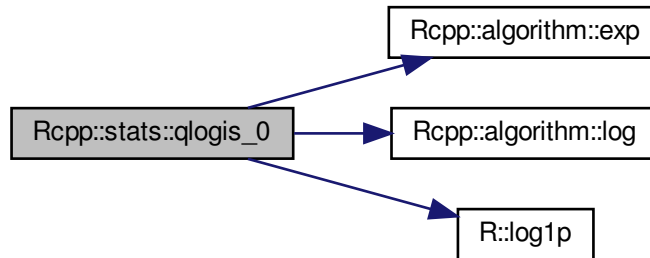
```
double Rcpp::stats::qlogis_0 (  
    double p,  
    int lower_tail,  
    int log_p ) [inline]
```

Definition at line 84 of file `logis.h`.



References Rcpp::algorithm::exp(), Rcpp::algorithm::log(), R::log1p(), ML\_NEGINF, ML\_POSINF, and R\_Q\_P01\_↔ boundaries.

Here is the call graph for this function:



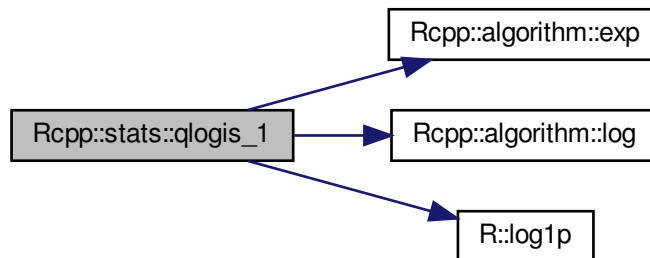
#### 5.9.1.34 qlogis\_1()

```
double Rcpp::stats::qlogis_1 (  
    double p,  
    double location,  
    int lower_tail,  
    int log_p ) [inline]
```

Definition at line 106 of file logis.h.

References Rcpp::algorithm::exp(), Rcpp::algorithm::log(), R::log1p(), ML\_NEGINF, ML\_POSINF, and R\_Q\_P01\_↔ boundaries.

Here is the call graph for this function:



### 5.9.1.35 `qnorm_0()`

```
double Rcpp::stats::qnorm_0 (  
    double p,  
    int lower_tail,  
    int log_p ) [inline]
```

Definition at line 103 of file `norm.h`.

### 5.9.1.36 `qnorm_1()`

```
double Rcpp::stats::qnorm_1 (  
    double p,  
    double mu,  
    int lower_tail,  
    int log_p ) [inline]
```

Definition at line 98 of file `norm.h`.

### 5.9.1.37 `qunif_0()`

```
double Rcpp::stats::qunif_0 (  
    double p,  
    int lower_tail,  
    int log_p ) [inline]
```

Definition at line 79 of file `unif.h`.

References `R_DT_qlv`, and `R_Q_P01_check`.

### 5.9.1.38 `qunif_1()`

```
double Rcpp::stats::qunif_1 (  
    double p,  
    double a,  
    int lower_tail,  
    int log_p ) [inline]
```

Definition at line 65 of file `unif.h`.

References `R_DT_qlv`, and `R_Q_P01_check`.

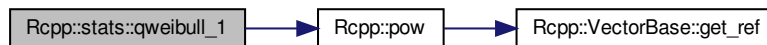
### 5.9.1.39 qweibull\_1()

```
double Rcpp::stats::qweibull_1 (
    double p,
    double shape,
    int lower_tail,
    int log_p ) [inline]
```

Definition at line 67 of file weibull.h.

References [ML\\_POSINF](#), [Rcpp::pow\(\)](#), [R\\_DT\\_Clog](#), and [R\\_Q\\_P01\\_boundaries](#).

Here is the call graph for this function:



## 5.10 Rcpp::sugar Namespace Reference

### Namespaces

- [cbind\\_impl](#)
- [detail](#)
- [median\\_detail](#)

### Classes

- class [IndexHash](#)
- class [SelfHash](#)
- class [SugarBlock\\_1](#)
- class [SugarBlock\\_2](#)
- class [SugarBlock\\_2\\_\\_VP](#)
- class [SugarBlock\\_2\\_\\_PV](#)
- class [SugarBlock\\_3\\_VVV](#)
- class [SugarMath\\_1](#)
- class [SugarMath\\_1< NA, RESULT\\_TYPE, int, T1, FunPtr >](#)
- class [SugarMath\\_1< false, RESULT\\_TYPE, int, T1, FunPtr >](#)
- class [Vectorized](#)
- class [Vectorized\\_INTSXP](#)
- class [Vectorized\\_INTSXP< Func, false, VEC >](#)
- class [All](#)
- class [All< false, T >](#)
- class [Any](#)

- class [Any< false, T >](#)
- struct [clamp\\_operator](#)
- struct [clamp\\_operator< REALSXP, true >](#)
- class [Clamp\\_Primitive\\_Vector\\_Primitive](#)
- class [SugarComplex](#)
- class [Cummax](#)
- class [Cummin](#)
- class [Cumprod](#)
- class [Cumsum](#)
- class [Diff](#)
- class [Diff< REALSXP, LHS\\_NA, LHS\\_T >](#)
- class [Diff< RTYPE, false, LHS\\_T >](#)
- class [Head](#)
- class [IfElse](#)
- class [IfElse< RTYPE, false, COND\\_T, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [IfElse\\_Primitive\\_Vector](#)
- class [IfElse\\_Primitive\\_Vector< RTYPE, false, COND\\_T, RHS\\_NA, RHS\\_T >](#)
- class [IfElse\\_Vector\\_Primitive](#)
- class [IfElse\\_Vector\\_Primitive< RTYPE, false, COND\\_T, LHS\\_NA, LHS\\_T >](#)
- class [IfElse\\_Primitive\\_Primitive](#)
- class [IfElse\\_Primitive\\_Primitive< RTYPE, false, COND\\_T >](#)
- class [IsFinite](#)
- class [IsInfinite](#)
- class [IsNa](#)
- class [IsNa< RTYPE, false, VEC\\_TYPE >](#)
- class [IsNa\\_Vector\\_is\\_na](#)
- class [IsNaN](#)
- class [Lapply](#)
- class [Lazy](#)
- class [Mapply\\_2](#)
- class [Mapply\\_2\\_Vector\\_Primitive](#)
- class [Mapply\\_2\\_Primitive\\_Vector](#)
- class [Mapply\\_3](#)
- class [Max](#)
- class [Max< RTYPE, false, T >](#)
- class [Mean](#)
- class [Mean< CPLXSP, NA, T >](#)
- class [Mean< LGLSP, NA, T >](#)
- class [Mean< INTSP, NA, T >](#)
- class [Median](#)
- class [Median< RTYPE, NA, T, true >](#)
- class [Median< RTYPE, false, T, NA\\_RM >](#)
- class [Median< STRSP, NA, T, NA\\_RM >](#)
- class [Median< STRSP, NA, T, true >](#)
- class [Median< STRSP, false, T, true >](#)
- class [Min](#)
- class [Min< RTYPE, false, T >](#)
- struct [pmax\\_op](#)
- struct [pmax\\_op< REALSXP, true, true >](#)
- struct [pmax\\_op< REALSXP, true, false >](#)
- struct [pmax\\_op< REALSXP, false, true >](#)

- struct [pmax\\_op](#)< REALSXP, false, false >
- struct [pmax\\_op](#)< INTSXP, LHS\_NA, RHS\_NA >
- class [pmax\\_op\\_Vector\\_Primitive](#)
- class [pmax\\_op\\_Vector\\_Primitive](#)< REALSXP, true >
- class [Pmax\\_Vector\\_Vector](#)
- class [Pmax\\_Vector\\_Primitive](#)
- struct [pmin\\_op](#)
- struct [pmin\\_op](#)< REALSXP, true, true >
- struct [pmin\\_op](#)< REALSXP, true, false >
- struct [pmin\\_op](#)< REALSXP, false, true >
- struct [pmin\\_op](#)< REALSXP, false, false >
- struct [pmin\\_op](#)< INTSXP, LHS\_NA, RHS\_NA >
- class [pmin\\_op\\_Vector\\_Primitive](#)
- class [pmin\\_op\\_Vector\\_Primitive](#)< REALSXP, true >
- class [Pmin\\_Vector\\_Vector](#)
- class [Pmin\\_Vector\\_Primitive](#)
- class [Pow](#)
- class [Pow](#)< INTSXP, NA, T, EXPONENT\_TYPE >
- class [Pow](#)< INTSXP, false, T, EXPONENT\_TYPE >
- class [Range](#)
- class [Range](#)< RTYPE, false, T >
- class [Rep](#)
- class [Rep\\_Single](#)
- class [Rep\\_each](#)
- class [Rep\\_len](#)
- class [Rev](#)
- class [RowSumsImpl](#)
- class [RowSumsImpl](#)< RTYPE, NA, T, true >
- class [RowSumsImpl](#)< RTYPE, false, T, NA\_RM >
- class [ColSumsImpl](#)
- class [ColSumsImpl](#)< RTYPE, NA, T, true >
- class [ColSumsImpl](#)< RTYPE, false, T, NA\_RM >
- class [RowMeansImpl](#)
- class [RowMeansImpl](#)< RTYPE, NA, T, true >
- class [RowMeansImpl](#)< RTYPE, false, T, NA\_RM >
- class [ColMeansImpl](#)
- class [ColMeansImpl](#)< RTYPE, NA, T, true >
- class [ColMeansImpl](#)< RTYPE, false, T, NA\_RM >
- struct [supply\\_application\\_result\\_of](#)
- class [Supply](#)
- class [Supply](#)< RTYPE, NA, T, Function, true >
- class [Sd](#)
- class [SelfInserter](#)
- class [SelfMatch](#)
- class [SeqLen](#)
- class [RemoveFromSet](#)
- class [SetDiff](#)
- class [SetEqual](#)
- class [Intersect](#)
- class [Union](#)
- class [sign\\_\\_impl](#)

- class [sign\\_\\_impl](#)< false, RTYPE >
- class [Sign](#)
- class [Sum](#)
- class [Sum](#)< REALSXP, NA, T >
- class [Sum](#)< RTYPE, false, T >
- class [CountInserter](#)
- class [Grabber](#)
- class [Table](#)
- class [Tail](#)
- class [InSet](#)
- class [In](#)
- class [Var](#)
- class [Var](#)< CPLXSXP, NA, T >
- class [WhichMax](#)
- class [WhichMax](#)< RTYPE, false, T >
- class [WhichMin](#)
- class [WhichMin](#)< RTYPE, false, T >
- class [And\\_SingleLogicalResult\\_SingleLogicalResult](#)
- class [And\\_SingleLogicalResult\\_SingleLogicalResult](#)< LHS\_NA, LHS\_T, false, RHS\_T >
- class [And\\_SingleLogicalResult\\_SingleLogicalResult](#)< false, LHS\_T, RHS\_NA, RHS\_T >
- class [And\\_SingleLogicalResult\\_SingleLogicalResult](#)< false, LHS\_T, false, RHS\_T >
- class [And\\_SingleLogicalResult\\_bool](#)
- class [And\\_LogicalExpression\\_LogicalExpression](#)
- class [And\\_LogicalExpression\\_LogicalExpression](#)< false, LHS\_T, RHS\_NA, RHS\_T >
- class [And\\_LogicalExpression\\_LogicalExpression](#)< LHS\_NA, LHS\_T, false, RHS\_T >
- class [And\\_LogicalExpression\\_LogicalExpression](#)< false, LHS\_T, false, RHS\_T >
- struct [negate](#)
- struct [negate](#)< false >
- class [Negate\\_SingleLogicalResult](#)
- class [Or\\_SingleLogicalResult\\_SingleLogicalResult](#)
- class [Or\\_SingleLogicalResult\\_SingleLogicalResult](#)< LHS\_NA, LHS\_T, false, RHS\_T >
- class [Or\\_SingleLogicalResult\\_SingleLogicalResult](#)< false, LHS\_T, RHS\_NA, RHS\_T >
- class [Or\\_SingleLogicalResult\\_SingleLogicalResult](#)< false, LHS\_T, false, RHS\_T >
- class [Or\\_SingleLogicalResult\\_bool](#)
- class [Or\\_LogicalExpression\\_LogicalExpression](#)
- class [Or\\_LogicalExpression\\_LogicalExpression](#)< false, LHS\_T, RHS\_NA, RHS\_T >
- class [Or\\_LogicalExpression\\_LogicalExpression](#)< LHS\_NA, LHS\_T, false, RHS\_T >
- class [Or\\_LogicalExpression\\_LogicalExpression](#)< false, LHS\_T, false, RHS\_T >
- class [forbidden\\_conversion](#)
- class [forbidden\\_conversion](#)< true >
- class [conversion\\_to\\_bool\\_is\\_forbidden](#)
- class [SingleLogicalResult](#)
- class [Col](#)
- class [Diag\\_Extractor](#)
- class [Diag\\_Maker](#)
- struct [diag\\_result\\_type\\_trait](#)
- class [LowerTri](#)
- class [Outer](#)
- class [Row](#)
- class [UpperTri](#)
- class [Nona](#)

- class [Nona](#)< RTYPE, NA, Rcpp::Vector< RTYPE > >
- class [NonaPrimitive](#)
- class [Comparator](#)
- class [Comparator](#)< RTYPE, Operator, false, LHS\_T, RHS\_NA, RHS\_T >
- class [Comparator](#)< RTYPE, Operator, false, LHS\_T, false, RHS\_T >
- class [Comparator\\_With\\_One\\_Value](#)
- class [Comparator\\_With\\_One\\_Value](#)< RTYPE, Operator, false, T >
- class [Divides\\_Vector\\_Vector](#)
- class [Divides\\_Vector\\_Vector](#)< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >
- class [Divides\\_Vector\\_Vector](#)< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >
- class [Divides\\_Vector\\_Vector](#)< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >
- class [Divides\\_Vector\\_Vector](#)< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >
- class [Divides\\_Vector\\_Vector](#)< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >
- class [Divides\\_Vector\\_Vector](#)< RTYPE, false, LHS\_T, false, RHS\_T >
- class [Divides\\_Vector\\_Vector](#)< REALSXP, false, LHS\_T, false, RHS\_T >
- class [Divides\\_Vector\\_Primitive](#)
- class [Divides\\_Vector\\_Primitive](#)< REALSXP, NA, T >
- class [Divides\\_Vector\\_Primitive](#)< RTYPE, false, T >
- class [Divides\\_Vector\\_Primitive](#)< REALSXP, false, T >
- class [Divides\\_Primitive\\_Vector](#)
- class [Divides\\_Primitive\\_Vector](#)< REALSXP, NA, T >
- class [Divides\\_Primitive\\_Vector](#)< RTYPE, false, T >
- class [Divides\\_Primitive\\_Vector](#)< REALSXP, false, T >
- class [Minus\\_Vector\\_Vector](#)
- class [Minus\\_Vector\\_Vector](#)< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >
- class [Minus\\_Vector\\_Vector](#)< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >
- class [Minus\\_Vector\\_Vector](#)< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >
- class [Minus\\_Vector\\_Vector](#)< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >
- class [Minus\\_Vector\\_Vector](#)< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >
- class [Minus\\_Vector\\_Vector](#)< RTYPE, false, LHS\_T, false, RHS\_T >
- class [Minus\\_Vector\\_Vector](#)< REALSXP, false, LHS\_T, false, RHS\_T >
- class [Minus\\_Vector\\_Primitive](#)
- class [Minus\\_Vector\\_Primitive](#)< REALSXP, NA, T >
- class [Minus\\_Vector\\_Primitive](#)< RTYPE, false, T >
- class [Minus\\_Vector\\_Primitive](#)< REALSXP, false, T >
- class [Minus\\_Primitive\\_Vector](#)
- class [Minus\\_Primitive\\_Vector](#)< REALSXP, NA, T >
- class [Minus\\_Primitive\\_Vector](#)< RTYPE, false, T >
- class [Minus\\_Primitive\\_Vector](#)< REALSXP, false, T >
- class [not\\_](#)
- class [not\\_](#)< RTYPE, false >
- class [not\\_](#)< REALSXP, NA >
- class [not\\_](#)< REALSXP, false >
- class [not\\_](#)< CPLXSXP, NA >
- class [not\\_](#)< CPLXSXP, false >
- class [Not\\_Vector](#)
- class [Plus\\_Vector\\_Vector](#)
- class [Plus\\_Vector\\_Vector](#)< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >
- class [Plus\\_Vector\\_Vector](#)< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >
- class [Plus\\_Vector\\_Vector](#)< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >
- class [Plus\\_Vector\\_Vector](#)< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >

- class [Plus\\_Vector\\_Vector](#)< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >
- class [Plus\\_Vector\\_Vector](#)< RTYPE, false, LHS\_T, false, RHS\_T >
- class [Plus\\_Vector\\_Vector](#)< REALSXP, false, LHS\_T, false, RHS\_T >
- class [Plus\\_Vector\\_Primitive](#)
- class [Plus\\_Vector\\_Primitive](#)< REALSXP, NA, T >
- class [Plus\\_Vector\\_Primitive](#)< RTYPE, false, T >
- class [Plus\\_Vector\\_Primitive](#)< REALSXP, false, T >
- class [Plus\\_Vector\\_Primitive\\_nona](#)
- class [Plus\\_Vector\\_Primitive\\_nona](#)< REALSXP, NA, T >
- class [Plus\\_Vector\\_Primitive\\_nona](#)< RTYPE, false, T >
- class [Plus\\_Vector\\_Primitive\\_nona](#)< REALSXP, false, T >
- class [Times\\_Vector\\_Vector](#)
- class [Times\\_Vector\\_Vector](#)< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >
- class [Times\\_Vector\\_Vector](#)< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >
- class [Times\\_Vector\\_Vector](#)< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >
- class [Times\\_Vector\\_Vector](#)< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >
- class [Times\\_Vector\\_Vector](#)< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >
- class [Times\\_Vector\\_Vector](#)< RTYPE, false, LHS\_T, false, RHS\_T >
- class [Times\\_Vector\\_Vector](#)< REALSXP, false, LHS\_T, false, RHS\_T >
- class [Times\\_Vector\\_Primitive](#)
- class [Times\\_Vector\\_Primitive](#)< REALSXP, NA, T >
- class [Times\\_Vector\\_Primitive](#)< RTYPE, false, T >
- class [Times\\_Vector\\_Primitive](#)< REALSXP, false, T >
- class [Times\\_Vector\\_Primitive\\_nona](#)
- class [Times\\_Vector\\_Primitive\\_nona](#)< REALSXP, NA, T >
- class [Times\\_Vector\\_Primitive\\_nona](#)< RTYPE, false, T >
- class [Times\\_Vector\\_Primitive\\_nona](#)< REALSXP, false, T >
- struct [unary\\_minus\\_result\\_type](#)
- struct [unary\\_minus\\_result\\_type](#)< LGLSXP >
- class [unary\\_minus](#)
- class [unary\\_minus](#)< RTYPE, false >
- class [unary\\_minus](#)< CPLXSXP, NA >
- class [unary\\_minus](#)< CPLXSXP, false >
- class [UnaryMinus\\_Vector](#)
- class [SugarIterator](#)
- struct [sugar\\_const\\_iterator\\_type](#)
- struct [sugar\\_const\\_iterator\\_type](#)< Rcpp::Vector< RTYPE > >
- struct [sugar\\_const\\_iterator\\_type](#)< CharacterVector >
- struct [is\\_sugar\\_vector](#)
- struct [is\\_sugar\\_vector](#)< Rcpp::Vector< RTYPE > >

## Typedefs

- typedef double(\* [DDFun](#)) (double)
- typedef [Nullable](#)< [Vector](#)< REALSXP > > [probs\\_t](#)



















T43 , typename T44 , typename T45 , typename T46 , typename T47 , typename T48 , typename T49 >

[cbind\\_impl::matrix\\_return](#)< T1 >::type [cbind](#) (const T1 &t1, const T2 &t2, const T3 &t3, const T4 &t4, const T5 &t5, const T6 &t6, const T7 &t7, const T8 &t8, const T9 &t9, const T10 &t10, const T11 &t11, const T12 &t12, const T13 &t13, const T14 &t14, const T15 &t15, const T16 &t16, const T17 &t17, const T18 &t18, const T19 &t19, const T20 &t20, const T21 &t21, const T22 &t22, const T23 &t23, const T24 &t24, const T25 &t25, const T26 &t26, const T27 &t27, const T28 &t28, const T29 &t29, const T30 &t30, const T31 &t31, const T32 &t32, const T33 &t33, const T34 &t34, const T35 &t35, const T36 &t36, const T37 &t37, const T38 &t38, const T39 &t39, const T40 &t40, const T41 &t41, const T42 &t42, const T43 &t43, const T44 &t44, const T45 &t45, const T46 &t46, const T47 &t47, const T48 &t48, const T49 &t49)

- [template](#)<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 , typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 , typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 , typename T45 , typename T46 , typename T47 , typename T48 , typename T49 , typename T50 >  
[cbind\\_impl::matrix\\_return](#)< T1 >::type [cbind](#) (const T1 &t1, const T2 &t2, const T3 &t3, const T4 &t4, const T5 &t5, const T6 &t6, const T7 &t7, const T8 &t8, const T9 &t9, const T10 &t10, const T11 &t11, const T12 &t12, const T13 &t13, const T14 &t14, const T15 &t15, const T16 &t16, const T17 &t17, const T18 &t18, const T19 &t19, const T20 &t20, const T21 &t21, const T22 &t22, const T23 &t23, const T24 &t24, const T25 &t25, const T26 &t26, const T27 &t27, const T28 &t28, const T29 &t29, const T30 &t30, const T31 &t31, const T32 &t32, const T33 &t33, const T34 &t34, const T35 &t35, const T36 &t36, const T37 &t37, const T38 &t38, const T39 &t39, const T40 &t40, const T41 &t41, const T42 &t42, const T43 &t43, const T44 &t44, const T45 &t45, const T46 &t46, const T47 &t47, const T48 &t48, const T49 &t49, const T50 &t50)
- [template](#)<int RTYPE, bool NA, typename T >  
[Vector](#)< RTYPE > [na\\_omit\\_impl](#) (const T &x, [Rcpp::traits::false\\_type](#))
- [template](#)<int RTYPE, bool NA, typename T >  
[Vector](#)< RTYPE > [na\\_omit\\_impl](#) (const T &x, [Rcpp::traits::true\\_type](#))
- [void](#) [Normalize](#) ([Vector](#)< REALSXP > &p, int require\_k, bool replace)
- [Vector](#)< INTSXP > [SampleReplace](#) ([Vector](#)< REALSXP > &p, int n, int k, bool one\_based)
- [template](#)<int RTYPE>  
[Vector](#)< RTYPE > [SampleReplace](#) ([Vector](#)< REALSXP > &p, int k, const [Vector](#)< RTYPE > &ref)
- [Vector](#)< INTSXP > [WalkerSample](#) (const [Vector](#)< REALSXP > &p, int n, int nans, bool one\_based)
- [template](#)<int RTYPE>  
[Vector](#)< RTYPE > [WalkerSample](#) (const [Vector](#)< REALSXP > &p, int nans, const [Vector](#)< RTYPE > &ref)
- [Vector](#)< INTSXP > [SampleNoReplace](#) ([Vector](#)< REALSXP > &p, int n, int nans, bool one\_based)
- [template](#)<int RTYPE>  
[Vector](#)< RTYPE > [SampleNoReplace](#) ([Vector](#)< REALSXP > &p, int nans, const [Vector](#)< RTYPE > &ref)
- [Vector](#)< INTSXP > [EmpiricalSample](#) (int n, int size, bool replace, bool one\_based)
- [template](#)<int RTYPE>  
[Vector](#)< RTYPE > [EmpiricalSample](#) (int size, bool replace, const [Vector](#)< RTYPE > &ref)
- [template](#)<typename Iterator >  
[String collapse\\_\\_impl](#) (Iterator it, R\_xlen\_t n)
- [template](#)<typename T >  
[sugar\\_const\\_iterator\\_type](#)< T >::type [get\\_const\\_begin\\_\\_impl](#) (const T &obj, [Rcpp::traits::true\\_type](#))
- [template](#)<typename T >  
[sugar\\_const\\_iterator\\_type](#)< T >::type [get\\_const\\_begin\\_\\_impl](#) (const T &obj, [Rcpp::traits::false\\_type](#))
- [template](#)<typename T >  
[sugar\\_const\\_iterator\\_type](#)< T >::type [get\\_const\\_begin](#) (const T &obj)
- [template](#)<> SEXP \* [get\\_const\\_begin](#) (const [CharacterVector](#) &obj)
- [template](#)<typename T >  
[sugar\\_const\\_iterator\\_type](#)< T >::type [get\\_const\\_end](#) (const T &obj)



## 5.10.1 Typedef Documentation

### 5.10.1.1 DDFun

```
typedef double(* Rcpp::sugar::DDFun) (double)
```

Definition at line 28 of file Vectorized\_Math.h.

### 5.10.1.2 probs\_t

```
typedef Nullable< Vector<REALSXP> > Rcpp::sugar::probs_t
```

Definition at line 395 of file sample.h.

## 5.10.2 Function Documentation

### 5.10.2.1 cbind() [1/49]

```
template<typename T1 , typename T2 >  
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (  
    const T1 & t1,  
    const T2 & t2 ) [inline]
```

Definition at line 516 of file cbind.h.

References MakeBindable.

### 5.10.2.2 cbind() [2/49]

```
template<typename T1 , typename T2 , typename T3 >  
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (  
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3 ) [inline]
```

Definition at line 522 of file cbind.h.

References MakeBindable.

### 5.10.2.3 cbind() [3/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4 ) [inline]
```

Definition at line 528 of file cbind.h.

References MakeBindable.

### 5.10.2.4 cbind() [4/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5 ) [inline]
```

Definition at line 535 of file cbind.h.

References MakeBindable.

### 5.10.2.5 cbind() [5/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6 ) [inline]
```

Definition at line 541 of file cbind.h.

References MakeBindable.

### 5.10.2.6 cbind() [6/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7 ) [inline]
```

Definition at line 547 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.7 cbind() [7/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8 ) [inline]
```

Definition at line 553 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.8 cbind() [8/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9 ) [inline]
```

Definition at line 559 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.9 cbind() [9/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10 ) [inline]
```

Definition at line 566 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.10 cbind() [10/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11 ) [inline]
```

Definition at line 572 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.11 cbind() [11/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12 ) [inline]
```

Definition at line 578 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.12 cbind() [12/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13 ) [inline]
```

Definition at line 584 of file cbind.h.

References [MakeBindable](#).

**5.10.2.13 cbind()** [13/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 >
```

```
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14 ) [inline]
```

Definition at line 590 of file cbind.h.

References [MakeBindable](#).

**5.10.2.14 cbind()** [14/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 >
```

```
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15 ) [inline]
```

Definition at line 597 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.15 cbind() [15/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16 ) [inline]
```

Definition at line 603 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.16 cbind() [16/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17 ) [inline]
```

Definition at line 609 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.17 `cbind()` [17/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18 ) [inline]
```

Definition at line 615 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.18 `cbind()` [18/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
```



```
const T9 & t9,  
const T10 & t10,  
const T11 & t11,  
const T12 & t12,  
const T13 & t13,  
const T14 & t14,  
const T15 & t15,  
const T16 & t16,  
const T17 & t17,  
const T18 & t18,  
const T19 & t19 ) [inline]
```

Definition at line 621 of file cbind.h.

References [MakeBindable](#).

#### 5.10.2.19 cbind() [19/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 >
```

```
cbind\_impl::matrix\_return<T1>::type Rcpp::sugar::cbind (  
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3,  
    const T4 & t4,  
    const T5 & t5,  
    const T6 & t6,  
    const T7 & t7,  
    const T8 & t8,  
    const T9 & t9,  
    const T10 & t10,  
    const T11 & t11,  
    const T12 & t12,  
    const T13 & t13,  
    const T14 & t14,  
    const T15 & t15,  
    const T16 & t16,  
    const T17 & t17,  
    const T18 & t18,  
    const T19 & t19,  
    const T20 & t20 ) [inline]
```

Definition at line 628 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.20 `cbind()` [20/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,
    const T21 & t21 ) [inline]
```

Definition at line 634 of file `cbind.h`.

References `MakeBindable`.

### 5.10.2.21 `cbind()` [21/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
```

```
const T11 & t11,  
const T12 & t12,  
const T13 & t13,  
const T14 & t14,  
const T15 & t15,  
const T16 & t16,  
const T17 & t17,  
const T18 & t18,  
const T19 & t19,  
const T20 & t20,  
const T21 & t21,  
const T22 & t22 ) [inline]
```

Definition at line 640 of file cbind.h.

References MakeBindable.

### 5.10.2.22 cbind() [22/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 , typename T21 , typename T22 , typename T23 >
```

```
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (  
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3,  
    const T4 & t4,  
    const T5 & t5,  
    const T6 & t6,  
    const T7 & t7,  
    const T8 & t8,  
    const T9 & t9,  
    const T10 & t10,  
    const T11 & t11,  
    const T12 & t12,  
    const T13 & t13,  
    const T14 & t14,  
    const T15 & t15,  
    const T16 & t16,  
    const T17 & t17,  
    const T18 & t18,  
    const T19 & t19,  
    const T20 & t20,  
    const T21 & t21,  
    const T22 & t22,  
    const T23 & t23 ) [inline]
```

Definition at line 646 of file cbind.h.

References MakeBindable.

**5.10.2.23 cbind() [23/49]**

```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,
    const T21 & t21,
    const T22 & t22,
    const T23 & t23,
    const T24 & t24 ) [inline]

```

Definition at line 652 of file cbind.h.

References [MakeBindable](#).

**5.10.2.24 cbind() [24/49]**

```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,

```

```
    const T8 & t8,  
    const T9 & t9,  
    const T10 & t10,  
    const T11 & t11,  
    const T12 & t12,  
    const T13 & t13,  
    const T14 & t14,  
    const T15 & t15,  
    const T16 & t16,  
    const T17 & t17,  
    const T18 & t18,  
    const T19 & t19,  
    const T20 & t20,  
    const T21 & t21,  
    const T22 & t22,  
    const T23 & t23,  
    const T24 & t24,  
    const T25 & t25 ) [inline]
```

Definition at line 659 of file cbind.h.

References [MakeBindable](#).

#### 5.10.2.25 `cbind()` [25/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename  
T26 >  
cbind\_impl::matrix\_return<T1>::type Rcpp::sugar::cbind (  
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3,  
    const T4 & t4,  
    const T5 & t5,  
    const T6 & t6,  
    const T7 & t7,  
    const T8 & t8,  
    const T9 & t9,  
    const T10 & t10,  
    const T11 & t11,  
    const T12 & t12,  
    const T13 & t13,  
    const T14 & t14,  
    const T15 & t15,  
    const T16 & t16,  
    const T17 & t17,  
    const T18 & t18,  
    const T19 & t19,
```

```
    const T20 & t20,  
    const T21 & t21,  
    const T22 & t22,  
    const T23 & t23,  
    const T24 & t24,  
    const T25 & t25,  
    const T26 & t26 ) [inline]
```

Definition at line 665 of file cbind.h.

References [MakeBindable](#).

#### 5.10.2.26 `cbind()` [26/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename  
T26 , typename T27 >
```

```
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (  
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3,  
    const T4 & t4,  
    const T5 & t5,  
    const T6 & t6,  
    const T7 & t7,  
    const T8 & t8,  
    const T9 & t9,  
    const T10 & t10,  
    const T11 & t11,  
    const T12 & t12,  
    const T13 & t13,  
    const T14 & t14,  
    const T15 & t15,  
    const T16 & t16,  
    const T17 & t17,  
    const T18 & t18,  
    const T19 & t19,  
    const T20 & t20,  
    const T21 & t21,  
    const T22 & t22,  
    const T23 & t23,  
    const T24 & t24,  
    const T25 & t25,  
    const T26 & t26,  
    const T27 & t27 ) [inline]
```

Definition at line 671 of file cbind.h.

References [MakeBindable](#).

**5.10.2.27 cbind() [27/49]**

```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,
    const T21 & t21,
    const T22 & t22,
    const T23 & t23,
    const T24 & t24,
    const T25 & t25,
    const T26 & t26,
    const T27 & t27,
    const T28 & t28 ) [inline]

```

Definition at line 677 of file cbind.h.

References MakeBindable.

**5.10.2.28 cbind() [28/49]**

```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,

```

```

const T2 & t2,
const T3 & t3,
const T4 & t4,
const T5 & t5,
const T6 & t6,
const T7 & t7,
const T8 & t8,
const T9 & t9,
const T10 & t10,
const T11 & t11,
const T12 & t12,
const T13 & t13,
const T14 & t14,
const T15 & t15,
const T16 & t16,
const T17 & t17,
const T18 & t18,
const T19 & t19,
const T20 & t20,
const T21 & t21,
const T22 & t22,
const T23 & t23,
const T24 & t24,
const T25 & t25,
const T26 & t26,
const T27 & t27,
const T28 & t28,
const T29 & t29 ) [inline]

```

Definition at line 683 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.29 `cbind()` [29/49]

```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,

```



```
const T10 & t10,  
const T11 & t11,  
const T12 & t12,  
const T13 & t13,  
const T14 & t14,  
const T15 & t15,  
const T16 & t16,  
const T17 & t17,  
const T18 & t18,  
const T19 & t19,  
const T20 & t20,  
const T21 & t21,  
const T22 & t22,  
const T23 & t23,  
const T24 & t24,  
const T25 & t25,  
const T26 & t26,  
const T27 & t27,  
const T28 & t28,  
const T29 & t29,  
const T30 & t30 ) [inline]
```

Definition at line 690 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.30 `cbind()` [30/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename  
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 >  
cbind\_impl::matrix\_return<T1>::type Rcpp::sugar::cbind (  
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3,  
    const T4 & t4,  
    const T5 & t5,  
    const T6 & t6,  
    const T7 & t7,  
    const T8 & t8,  
    const T9 & t9,  
    const T10 & t10,  
    const T11 & t11,  
    const T12 & t12,  
    const T13 & t13,  
    const T14 & t14,  
    const T15 & t15,  
    const T16 & t16,
```

```

    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,
    const T21 & t21,
    const T22 & t22,
    const T23 & t23,
    const T24 & t24,
    const T25 & t25,
    const T26 & t26,
    const T27 & t27,
    const T28 & t28,
    const T29 & t29,
    const T30 & t30,
    const T31 & t31 ) [inline]

```

Definition at line 696 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.31 `cbind()` [31/49]

```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,
    const T21 & t21,
    const T22 & t22,

```

```
    const T23 & t23,  
    const T24 & t24,  
    const T25 & t25,  
    const T26 & t26,  
    const T27 & t27,  
    const T28 & t28,  
    const T29 & t29,  
    const T30 & t30,  
    const T31 & t31,  
    const T32 & t32 ) [inline]
```

Definition at line 702 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.32 cbind() [32/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename  
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,  
typename T33 >  
cbind\_impl::matrix\_return<T1>::type Rcpp::sugar::cbind (  
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3,  
    const T4 & t4,  
    const T5 & t5,  
    const T6 & t6,  
    const T7 & t7,  
    const T8 & t8,  
    const T9 & t9,  
    const T10 & t10,  
    const T11 & t11,  
    const T12 & t12,  
    const T13 & t13,  
    const T14 & t14,  
    const T15 & t15,  
    const T16 & t16,  
    const T17 & t17,  
    const T18 & t18,  
    const T19 & t19,  
    const T20 & t20,  
    const T21 & t21,  
    const T22 & t22,  
    const T23 & t23,  
    const T24 & t24,  
    const T25 & t25,  
    const T26 & t26,
```

```
    const T27 & t27,  
    const T28 & t28,  
    const T29 & t29,  
    const T30 & t30,  
    const T31 & t31,  
    const T32 & t32,  
    const T33 & t33 ) [inline]
```

Definition at line 708 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.33 `cbind()` [33/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename  
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,  
typename T33 , typename T34 >  
cbind\_impl::matrix\_return<T1>::type Rcpp::sugar::cbind (  
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3,  
    const T4 & t4,  
    const T5 & t5,  
    const T6 & t6,  
    const T7 & t7,  
    const T8 & t8,  
    const T9 & t9,  
    const T10 & t10,  
    const T11 & t11,  
    const T12 & t12,  
    const T13 & t13,  
    const T14 & t14,  
    const T15 & t15,  
    const T16 & t16,  
    const T17 & t17,  
    const T18 & t18,  
    const T19 & t19,  
    const T20 & t20,  
    const T21 & t21,  
    const T22 & t22,  
    const T23 & t23,  
    const T24 & t24,  
    const T25 & t25,  
    const T26 & t26,  
    const T27 & t27,  
    const T28 & t28,  
    const T29 & t29,
```

```
    const T30 & t30,  
    const T31 & t31,  
    const T32 & t32,  
    const T33 & t33,  
    const T34 & t34 ) [inline]
```

Definition at line 714 of file cbind.h.

References MakeBindable.

### 5.10.2.34 cbind() [34/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename  
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,  
typename T33 , typename T34 , typename T35 >
```

```
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
```

```
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3,  
    const T4 & t4,  
    const T5 & t5,  
    const T6 & t6,  
    const T7 & t7,  
    const T8 & t8,  
    const T9 & t9,  
    const T10 & t10,  
    const T11 & t11,  
    const T12 & t12,  
    const T13 & t13,  
    const T14 & t14,  
    const T15 & t15,  
    const T16 & t16,  
    const T17 & t17,  
    const T18 & t18,  
    const T19 & t19,  
    const T20 & t20,  
    const T21 & t21,  
    const T22 & t22,  
    const T23 & t23,  
    const T24 & t24,  
    const T25 & t25,  
    const T26 & t26,  
    const T27 & t27,  
    const T28 & t28,  
    const T29 & t29,  
    const T30 & t30,  
    const T31 & t31,
```

```

    const T32 & t32,
    const T33 & t33,
    const T34 & t34,
    const T35 & t35 ) [inline]

```

Definition at line 721 of file cbind.h.

References MakeBindable.

### 5.10.2.35 cbind() [35/49]

```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (

```

```

    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,
    const T21 & t21,
    const T22 & t22,
    const T23 & t23,
    const T24 & t24,
    const T25 & t25,
    const T26 & t26,
    const T27 & t27,
    const T28 & t28,
    const T29 & t29,
    const T30 & t30,
    const T31 & t31,
    const T32 & t32,

```

```
    const T33 & t33,  
    const T34 & t34,  
    const T35 & t35,  
    const T36 & t36 ) [inline]
```

Definition at line 727 of file cbind.h.

References MakeBindable.

### 5.10.2.36 cbind() [36/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename  
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,  
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 >  
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
```

```
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3,  
    const T4 & t4,  
    const T5 & t5,  
    const T6 & t6,  
    const T7 & t7,  
    const T8 & t8,  
    const T9 & t9,  
    const T10 & t10,  
    const T11 & t11,  
    const T12 & t12,  
    const T13 & t13,  
    const T14 & t14,  
    const T15 & t15,  
    const T16 & t16,  
    const T17 & t17,  
    const T18 & t18,  
    const T19 & t19,  
    const T20 & t20,  
    const T21 & t21,  
    const T22 & t22,  
    const T23 & t23,  
    const T24 & t24,  
    const T25 & t25,  
    const T26 & t26,  
    const T27 & t27,  
    const T28 & t28,  
    const T29 & t29,  
    const T30 & t30,  
    const T31 & t31,  
    const T32 & t32,
```

```

    const T33 & t33,
    const T34 & t34,
    const T35 & t35,
    const T36 & t36,
    const T37 & t37 ) [inline]

```

Definition at line 733 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.37 cbind() [37/49]

```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,
    const T21 & t21,
    const T22 & t22,
    const T23 & t23,
    const T24 & t24,
    const T25 & t25,
    const T26 & t26,
    const T27 & t27,
    const T28 & t28,
    const T29 & t29,
    const T30 & t30,
    const T31 & t31,

```



```
    const T32 & t32,  
    const T33 & t33,  
    const T34 & t34,  
    const T35 & t35,  
    const T36 & t36,  
    const T37 & t37,  
    const T38 & t38 ) [inline]
```

Definition at line 739 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.38 cbind() [38/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename  
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,  
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename  
T39 >
```

```
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (  
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3,  
    const T4 & t4,  
    const T5 & t5,  
    const T6 & t6,  
    const T7 & t7,  
    const T8 & t8,  
    const T9 & t9,  
    const T10 & t10,  
    const T11 & t11,  
    const T12 & t12,  
    const T13 & t13,  
    const T14 & t14,  
    const T15 & t15,  
    const T16 & t16,  
    const T17 & t17,  
    const T18 & t18,  
    const T19 & t19,  
    const T20 & t20,  
    const T21 & t21,  
    const T22 & t22,  
    const T23 & t23,  
    const T24 & t24,  
    const T25 & t25,  
    const T26 & t26,  
    const T27 & t27,  
    const T28 & t28,
```

```

    const T29 & t29,
    const T30 & t30,
    const T31 & t31,
    const T32 & t32,
    const T33 & t33,
    const T34 & t34,
    const T35 & t35,
    const T36 & t36,
    const T37 & t37,
    const T38 & t38,
    const T39 & t39 ) [inline]

```

Definition at line 745 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.39 `cbind()` [39/49]

```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 >

```

```

cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,
    const T21 & t21,
    const T22 & t22,
    const T23 & t23,
    const T24 & t24,

```

```
    const T25 & t25,  
    const T26 & t26,  
    const T27 & t27,  
    const T28 & t28,  
    const T29 & t29,  
    const T30 & t30,  
    const T31 & t31,  
    const T32 & t32,  
    const T33 & t33,  
    const T34 & t34,  
    const T35 & t35,  
    const T36 & t36,  
    const T37 & t37,  
    const T38 & t38,  
    const T39 & t39,  
    const T40 & t40 ) [inline]
```

Definition at line 752 of file cbind.h.

References [MakeBindable](#).

#### 5.10.2.40 `cbind()` [40/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename  
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,  
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename  
T39 , typename T40 , typename T41 >  
cbind\_impl::matrix\_return<T1>::type Rcpp::sugar::cbind (  
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3,  
    const T4 & t4,  
    const T5 & t5,  
    const T6 & t6,  
    const T7 & t7,  
    const T8 & t8,  
    const T9 & t9,  
    const T10 & t10,  
    const T11 & t11,  
    const T12 & t12,  
    const T13 & t13,  
    const T14 & t14,  
    const T15 & t15,  
    const T16 & t16,  
    const T17 & t17,  
    const T18 & t18,  
    const T19 & t19,
```

```

const T20 & t20,
const T21 & t21,
const T22 & t22,
const T23 & t23,
const T24 & t24,
const T25 & t25,
const T26 & t26,
const T27 & t27,
const T28 & t28,
const T29 & t29,
const T30 & t30,
const T31 & t31,
const T32 & t32,
const T33 & t33,
const T34 & t34,
const T35 & t35,
const T36 & t36,
const T37 & t37,
const T38 & t38,
const T39 & t39,
const T40 & t40,
const T41 & t41 ) [inline]

```

Definition at line 758 of file cbind.h.

References [MakeBindable](#).

#### 5.10.2.41 `cbind()` [41/49]

```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,

```

```

    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,
    const T21 & t21,
    const T22 & t22,
    const T23 & t23,
    const T24 & t24,
    const T25 & t25,
    const T26 & t26,
    const T27 & t27,
    const T28 & t28,
    const T29 & t29,
    const T30 & t30,
    const T31 & t31,
    const T32 & t32,
    const T33 & t33,
    const T34 & t34,
    const T35 & t35,
    const T36 & t36,
    const T37 & t37,
    const T38 & t38,
    const T39 & t39,
    const T40 & t40,
    const T41 & t41,
    const T42 & t42 ) [inline]

```

Definition at line 764 of file cbind.h.

References [MakeBindable](#).

#### 5.10.2.42 [cbind\(\)](#) [42/49]

```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 , typename T43 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,

```

```
const T7 & t7,  
const T8 & t8,  
const T9 & t9,  
const T10 & t10,  
const T11 & t11,  
const T12 & t12,  
const T13 & t13,  
const T14 & t14,  
const T15 & t15,  
const T16 & t16,  
const T17 & t17,  
const T18 & t18,  
const T19 & t19,  
const T20 & t20,  
const T21 & t21,  
const T22 & t22,  
const T23 & t23,  
const T24 & t24,  
const T25 & t25,  
const T26 & t26,  
const T27 & t27,  
const T28 & t28,  
const T29 & t29,  
const T30 & t30,  
const T31 & t31,  
const T32 & t32,  
const T33 & t33,  
const T34 & t34,  
const T35 & t35,  
const T36 & t36,  
const T37 & t37,  
const T38 & t38,  
const T39 & t39,  
const T40 & t40,  
const T41 & t41,  
const T42 & t42,  
const T43 & t43 ) [inline]
```

Definition at line 770 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.43 `cbind()` [43/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename  
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,  
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
```

```
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,
    const T21 & t21,
    const T22 & t22,
    const T23 & t23,
    const T24 & t24,
    const T25 & t25,
    const T26 & t26,
    const T27 & t27,
    const T28 & t28,
    const T29 & t29,
    const T30 & t30,
    const T31 & t31,
    const T32 & t32,
    const T33 & t33,
    const T34 & t34,
    const T35 & t35,
    const T36 & t36,
    const T37 & t37,
    const T38 & t38,
    const T39 & t39,
    const T40 & t40,
    const T41 & t41,
    const T42 & t42,
    const T43 & t43,
    const T44 & t44 ) [inline]
```

Definition at line 776 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.44 cbind() [44/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 , typename T45 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,
    const T21 & t21,
    const T22 & t22,
    const T23 & t23,
    const T24 & t24,
    const T25 & t25,
    const T26 & t26,
    const T27 & t27,
    const T28 & t28,
    const T29 & t29,
    const T30 & t30,
    const T31 & t31,
    const T32 & t32,
    const T33 & t33,
    const T34 & t34,
    const T35 & t35,
    const T36 & t36,
    const T37 & t37,
    const T38 & t38,
    const T39 & t39,
    const T40 & t40,
    const T41 & t41,
    const T42 & t42,
    const T43 & t43,
```



```
    const T44 & t44,  
    const T45 & t45 ) [inline]
```

Definition at line 783 of file cbind.h.

References MakeBindable.

#### 5.10.2.45 cbind() [45/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename  
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,  
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename  
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 , typename T45 ,  
typename T46 >
```

```
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (  
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3,  
    const T4 & t4,  
    const T5 & t5,  
    const T6 & t6,  
    const T7 & t7,  
    const T8 & t8,  
    const T9 & t9,  
    const T10 & t10,  
    const T11 & t11,  
    const T12 & t12,  
    const T13 & t13,  
    const T14 & t14,  
    const T15 & t15,  
    const T16 & t16,  
    const T17 & t17,  
    const T18 & t18,  
    const T19 & t19,  
    const T20 & t20,  
    const T21 & t21,  
    const T22 & t22,  
    const T23 & t23,  
    const T24 & t24,  
    const T25 & t25,  
    const T26 & t26,  
    const T27 & t27,  
    const T28 & t28,  
    const T29 & t29,  
    const T30 & t30,  
    const T31 & t31,  
    const T32 & t32,
```

```

    const T33 & t33,
    const T34 & t34,
    const T35 & t35,
    const T36 & t36,
    const T37 & t37,
    const T38 & t38,
    const T39 & t39,
    const T40 & t40,
    const T41 & t41,
    const T42 & t42,
    const T43 & t43,
    const T44 & t44,
    const T45 & t45,
    const T46 & t46 ) [inline]

```

Definition at line 789 of file cbind.h.

References [MakeBindable](#).

#### 5.10.2.46 `cbind()` [46/49]

```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 , typename T45 ,
typename T46 , typename T47 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,

```

```
    const T21 & t21,  
    const T22 & t22,  
    const T23 & t23,  
    const T24 & t24,  
    const T25 & t25,  
    const T26 & t26,  
    const T27 & t27,  
    const T28 & t28,  
    const T29 & t29,  
    const T30 & t30,  
    const T31 & t31,  
    const T32 & t32,  
    const T33 & t33,  
    const T34 & t34,  
    const T35 & t35,  
    const T36 & t36,  
    const T37 & t37,  
    const T38 & t38,  
    const T39 & t39,  
    const T40 & t40,  
    const T41 & t41,  
    const T42 & t42,  
    const T43 & t43,  
    const T44 & t44,  
    const T45 & t45,  
    const T46 & t46,  
    const T47 & t47 ) [inline]
```

Definition at line 795 of file cbind.h.

References [MakeBindable](#).

#### 5.10.2.47 `cbind()` [47/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename  
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,  
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename  
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,  
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename  
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 , typename T45 ,  
typename T46 , typename T47 , typename T48 >  
cbind\_impl::matrix\_return<T1>::type Rcpp::sugar::cbind (  
    const T1 & t1,  
    const T2 & t2,  
    const T3 & t3,  
    const T4 & t4,  
    const T5 & t5,  
    const T6 & t6,  
    const T7 & t7,
```

```
const T8 & t8,  
const T9 & t9,  
const T10 & t10,  
const T11 & t11,  
const T12 & t12,  
const T13 & t13,  
const T14 & t14,  
const T15 & t15,  
const T16 & t16,  
const T17 & t17,  
const T18 & t18,  
const T19 & t19,  
const T20 & t20,  
const T21 & t21,  
const T22 & t22,  
const T23 & t23,  
const T24 & t24,  
const T25 & t25,  
const T26 & t26,  
const T27 & t27,  
const T28 & t28,  
const T29 & t29,  
const T30 & t30,  
const T31 & t31,  
const T32 & t32,  
const T33 & t33,  
const T34 & t34,  
const T35 & t35,  
const T36 & t36,  
const T37 & t37,  
const T38 & t38,  
const T39 & t39,  
const T40 & t40,  
const T41 & t41,  
const T42 & t42,  
const T43 & t43,  
const T44 & t44,  
const T45 & t45,  
const T46 & t46,  
const T47 & t47,  
const T48 & t48 ) [inline]
```

Definition at line 801 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.48 `cbind()` [48/49]

```
template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,  
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
```

```
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 , typename T45 ,
typename T46 , typename T47 , typename T48 , typename T49 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (
    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,
    const T21 & t21,
    const T22 & t22,
    const T23 & t23,
    const T24 & t24,
    const T25 & t25,
    const T26 & t26,
    const T27 & t27,
    const T28 & t28,
    const T29 & t29,
    const T30 & t30,
    const T31 & t31,
    const T32 & t32,
    const T33 & t33,
    const T34 & t34,
    const T35 & t35,
    const T36 & t36,
    const T37 & t37,
    const T38 & t38,
    const T39 & t39,
    const T40 & t40,
    const T41 & t41,
    const T42 & t42,
    const T43 & t43,
    const T44 & t44,
    const T45 & t45,
    const T46 & t46,
    const T47 & t47,
```

```

    const T48 & t48,
    const T49 & t49 ) [inline]

```

Definition at line 807 of file cbind.h.

References [MakeBindable](#).

### 5.10.2.49 cbind() [49/49]

```

template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 ,
typename T7 , typename T8 , typename T9 , typename T10 , typename T11 , typename T12 , typename
T13 , typename T14 , typename T15 , typename T16 , typename T17 , typename T18 , typename T19 ,
typename T20 , typename T21 , typename T22 , typename T23 , typename T24 , typename T25 , typename
T26 , typename T27 , typename T28 , typename T29 , typename T30 , typename T31 , typename T32 ,
typename T33 , typename T34 , typename T35 , typename T36 , typename T37 , typename T38 , typename
T39 , typename T40 , typename T41 , typename T42 , typename T43 , typename T44 , typename T45 ,
typename T46 , typename T47 , typename T48 , typename T49 , typename T50 >
cbind_impl::matrix_return<T1>::type Rcpp::sugar::cbind (

```

```

    const T1 & t1,
    const T2 & t2,
    const T3 & t3,
    const T4 & t4,
    const T5 & t5,
    const T6 & t6,
    const T7 & t7,
    const T8 & t8,
    const T9 & t9,
    const T10 & t10,
    const T11 & t11,
    const T12 & t12,
    const T13 & t13,
    const T14 & t14,
    const T15 & t15,
    const T16 & t16,
    const T17 & t17,
    const T18 & t18,
    const T19 & t19,
    const T20 & t20,
    const T21 & t21,
    const T22 & t22,
    const T23 & t23,
    const T24 & t24,
    const T25 & t25,
    const T26 & t26,
    const T27 & t27,
    const T28 & t28,
    const T29 & t29,
    const T30 & t30,
    const T31 & t31,
    const T32 & t32,

```

```
const T33 & t33,  
const T34 & t34,  
const T35 & t35,  
const T36 & t36,  
const T37 & t37,  
const T38 & t38,  
const T39 & t39,  
const T40 & t40,  
const T41 & t41,  
const T42 & t42,  
const T43 & t43,  
const T44 & t44,  
const T45 & t45,  
const T46 & t46,  
const T47 & t47,  
const T48 & t48,  
const T49 & t49,  
const T50 & t50 ) [inline]
```

Definition at line 814 of file cbind.h.

References MakeBindable.

#### 5.10.2.50 collapse\_\_impl()

```
template<typename Iterator >  
String Rcpp::sugar::collapse__impl (  
    Iterator it,  
    R_xlen_t n ) [inline]
```

Definition at line 28 of file collapse.h.

Referenced by Rcpp::collapse().

#### 5.10.2.51 EmpiricalSample() [1/2]

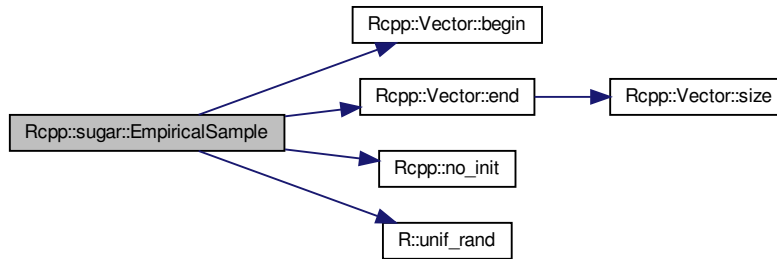
```
Vector<INTSXP> Rcpp::sugar::EmpiricalSample (  
    int n,  
    int size,  
    bool replace,  
    bool one_based ) [inline]
```

Definition at line 337 of file sample.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp←  
::no\_init(), and R::unif\_rand().

Referenced by `Rcpp::sample()`.

Here is the call graph for this function:



### 5.10.2.52 EmpiricalSample() [2/2]

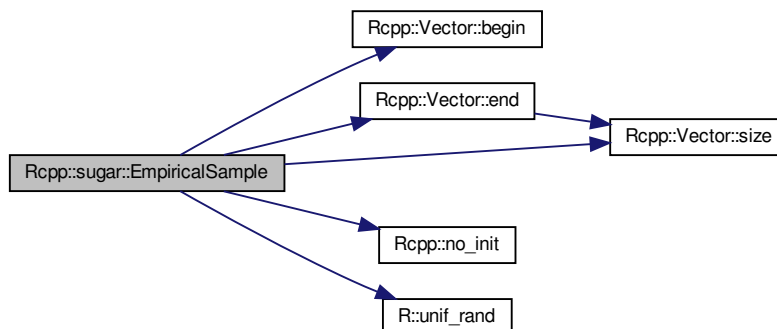
```

template<int RTYPE>
Vector<RTYPE> Rcpp::sugar::EmpiricalSample (
    int size,
    bool replace,
    const Vector< RTYPE > & ref ) [inline]
  
```

Definition at line 367 of file `sample.h`.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::no_init()`, `Rcpp::Vector< RTYPE, StoragePolicy >::size()`, and `R::unif_rand()`.

Here is the call graph for this function:





**5.10.2.53** `get_const_begin()` [1/2]

```
template<>
SEXP* Rcpp::sugar::get_const_begin (
    const CharacterVector & obj ) [inline]
```

Definition at line 143 of file iterator.h.

References `get_string_ptr()`.

Here is the call graph for this function:

**5.10.2.54** `get_const_begin()` [2/2]

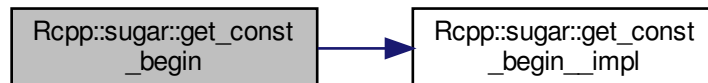
```
template<typename T >
sugar\_const\_iterator\_type<T>::type Rcpp::sugar::get_const_begin (
    const T & obj ) [inline]
```

Definition at line 138 of file iterator.h.

References `get_const_begin__impl()`.

Referenced by `Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Intersect()`, and `Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Union()`.

Here is the call graph for this function:



### 5.10.2.55 `get_const_begin_impl()` [1/2]

```
template<typename T >
sugar_const_iterator_type<T>::type Rcpp::sugar::get_const_begin_impl (
    const T & obj,
    Rcpp::traits::false_type ) [inline]
```

Definition at line 130 of file iterator.h.

### 5.10.2.56 `get_const_begin_impl()` [2/2]

```
template<typename T >
sugar_const_iterator_type<T>::type Rcpp::sugar::get_const_begin_impl (
    const T & obj,
    Rcpp::traits::true_type ) [inline]
```

Definition at line 126 of file iterator.h.

Referenced by `get_const_begin()`.

### 5.10.2.57 `get_const_end()`

```
template<typename T >
sugar_const_iterator_type<T>::type Rcpp::sugar::get_const_end (
    const T & obj ) [inline]
```

Definition at line 148 of file iterator.h.

Referenced by `Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Intersect()`, and `Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Union()`.

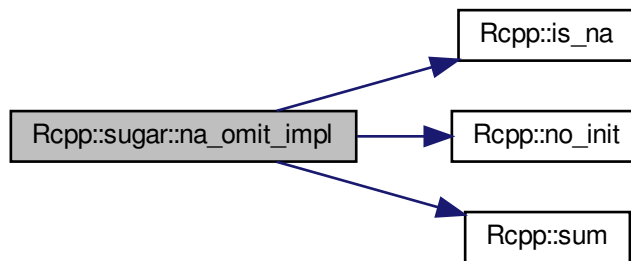
**5.10.2.58 na\_omit\_impl() [1/2]**

```
template<int RTYPE, bool NA, typename T >  
Vector<RTYPE> Rcpp::sugar::na_omit_impl (  
    const T & x,  
    Rcpp::traits::false_type )
```

Definition at line 29 of file na\_omit.h.

References Rcpp::is\_na(), Rcpp::no\_init(), and Rcpp::sum().

Here is the call graph for this function:

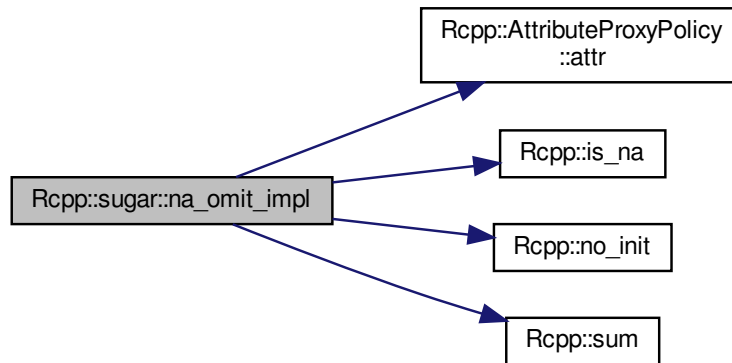
**5.10.2.59 na\_omit\_impl() [2/2]**

```
template<int RTYPE, bool NA, typename T >  
Vector<RTYPE> Rcpp::sugar::na_omit_impl (  
    const T & x,  
    Rcpp::traits::true_type )
```

Definition at line 44 of file na\_omit.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::is\_na(), Rcpp::no\_init(), and Rcpp::sum().

Here is the call graph for this function:



### 5.10.2.60 Normalize()

```

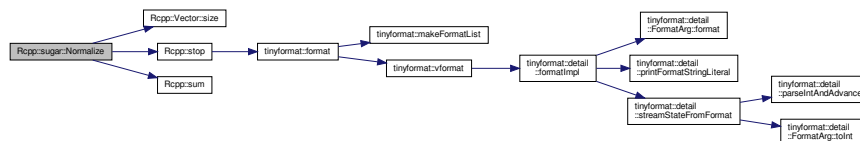
void Rcpp::sugar::Normalize (
    Vector< REALSXP > & p,
    int require_k,
    bool replace ) [inline]
  
```

Definition at line 57 of file `sample.h`.

References `Rcpp::Vector< RTYPE, StoragePolicy >::size()`, `Rcpp::stop()`, and `Rcpp::sum()`.

Referenced by `Rcpp::sample()`.

Here is the call graph for this function:



## 5.10.2.61 SampleNoReplace() [1/2]

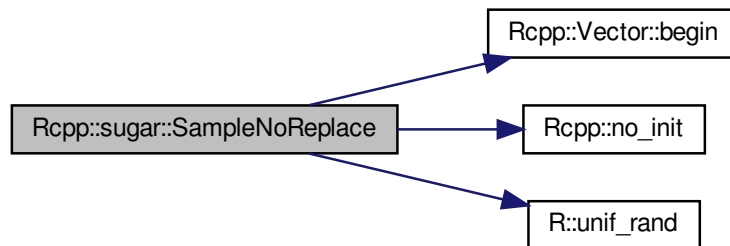
```
Vector<INTSXP> Rcpp::sugar::SampleNoReplace (
    Vector< REALSXP > & p,
    int n,
    int nans,
    bool one_based ) [inline]
```

Definition at line 255 of file sample.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::no\_init(), and R::unif\_rand().

Referenced by Rcpp::sample().

Here is the call graph for this function:



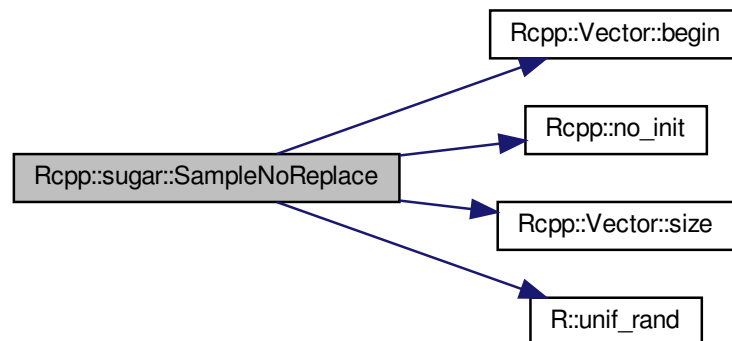
## 5.10.2.62 SampleNoReplace() [2/2]

```
template<int RTYPE>
Vector<RTYPE> Rcpp::sugar::SampleNoReplace (
    Vector< REALSXP > & p,
    int nans,
    const Vector< RTYPE > & ref ) [inline]
```

Definition at line 295 of file sample.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::no\_init(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), and R::unif\_rand().

Here is the call graph for this function:



### 5.10.2.63 SampleReplace() [1/2]

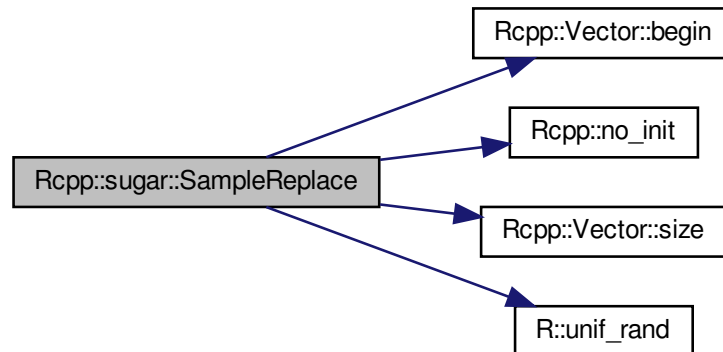
```

template<int RTYPE>
Vector<RTYPE> Rcpp::sugar::SampleReplace (
    Vector< REALSXP > & p,
    int k,
    const Vector< RTYPE > & ref ) [inline]
  
```

Definition at line 114 of file `sample.h`.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::no_init()`, `Rcpp::Vector< RTYPE, StoragePolicy >::size()`, and `R::unif_rand()`.

Here is the call graph for this function:



#### 5.10.2.64 SampleReplace() [2/2]

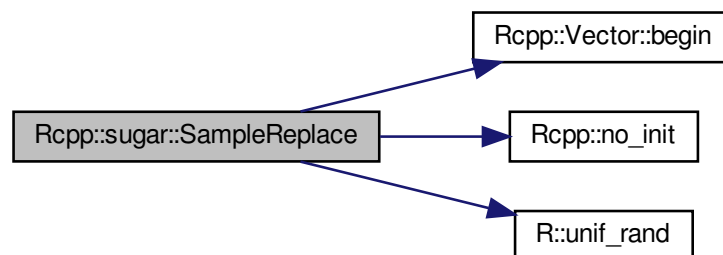
```
Vector<INTSXP> Rcpp::sugar::SampleReplace (  
    Vector< REALSXP > & p,  
    int n,  
    int k,  
    bool one_based ) [inline]
```

Definition at line 81 of file `sample.h`.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::no_init()`, and `R::unif_rand()`.

Referenced by `Rcpp::sample()`.

Here is the call graph for this function:



**5.10.2.65 WalkerSample() [1/2]**

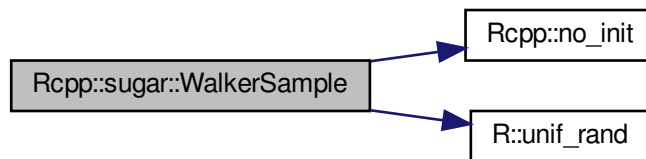
```
Vector<INTSXP> Rcpp::sugar::WalkerSample (
    const Vector< REALSXP > & p,
    int n,
    int nans,
    bool one_based ) [inline]
```

Definition at line 149 of file sample.h.

References Rcpp::no\_init(), and R::unif\_rand().

Referenced by Rcpp::sample().

Here is the call graph for this function:

**5.10.2.66 WalkerSample() [2/2]**

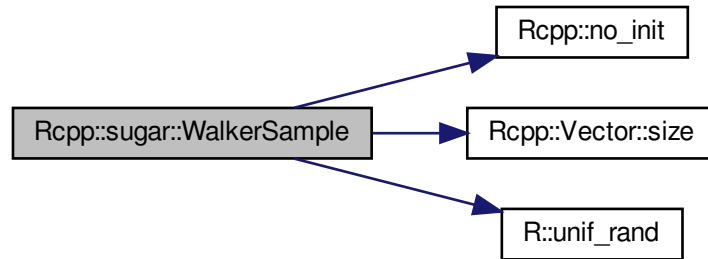
```
template<int RTYPE>
Vector<RTYPE> Rcpp::sugar::WalkerSample (
    const Vector< REALSXP > & p,
    int nans,
    const Vector< RTYPE > & ref ) [inline]
```

Definition at line 201 of file sample.h.

References Rcpp::no\_init(), Rcpp::Vector< RTYPE, StoragePolicy >::size(), and R::unif\_rand().



Here is the call graph for this function:



## 5.11 Rcpp::sugar::cbind\_impl Namespace Reference

### Namespaces

- [detail](#)

### Classes

- struct [cbind\\_sexptype\\_traits](#)
- struct [cbind\\_sexptype\\_traits< SEXP >](#)
- struct [cbind\\_storage\\_type](#)
- struct [cbind\\_storage\\_type< LGLSXP >](#)
- class [BindableExpression](#)
- class [ContainerBindable](#)
- struct [scalar](#)
- class [ScalarBindable](#)
- class [JoinOp](#)
- class [JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >](#)
- class [JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >](#)
- class [JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type >](#)
- struct [matrix\\_return](#)
- struct [matrix\\_return< T, false >](#)

## Functions

- `template<int RTYPE>`  
[ContainerBindable](#)< RTYPE, [Rcpp::Matrix](#)< RTYPE > > [MakeContainerBindable](#) (const [Rcpp::Matrix](#)< RTYPE > &x)
- `template<int RTYPE>`  
[ContainerBindable](#)< RTYPE, [Rcpp::Vector](#)< RTYPE > > [MakeContainerBindable](#) (const [Rcpp::Vector](#)< RTYPE > &x)
- `template<>` [ContainerBindable](#)< LGLSXP, [Rcpp::Matrix](#)< LGLSXP > > [MakeContainerBindable](#) (const [Rcpp::Matrix](#)< LGLSXP > &x)
- `template<>` [ContainerBindable](#)< LGLSXP, [Rcpp::Vector](#)< LGLSXP > > [MakeContainerBindable](#) (const [Rcpp::Vector](#)< LGLSXP > &x)
- `template<typename T >`  
[ScalarBindable](#)< T > [MakeScalarBindable](#) (const T &t)
- `template<int RTYPE, typename E1 , typename E2 >`  
[JoinOp](#)< RTYPE, E1, E2 > [operator](#), (const [BindableExpression](#)< RTYPE, E1 > &e1, const [BindableExpression](#)< RTYPE, E2 > &e2)

### 5.11.1 Function Documentation

#### 5.11.1.1 MakeContainerBindable() [1/4]

```
template<>
ContainerBindable<LGLSXP, Rcpp::Matrix<LGLSXP> > Rcpp::sugar::cbind_impl::MakeContainerBindable (
    (
        const Rcpp::Matrix< LGLSXP > & x ) [inline]
```

Definition at line 343 of file cbind.h.

#### 5.11.1.2 MakeContainerBindable() [2/4]

```
template<int RTYPE>
ContainerBindable<RTYPE, Rcpp::Matrix<RTYPE> > Rcpp::sugar::cbind_impl::MakeContainerBindable (
    const Rcpp::Matrix< RTYPE > & x ) [inline]
```

Definition at line 331 of file cbind.h.

Referenced by [Rcpp::sugar::cbind\\_impl::detail::MakeBindableCall](#)< T, true >::operator(), [Rcpp::sugar::cbind\\_impl::detail::MakeBindableCall](#)< [Rcpp::Matrix](#)< LGLSXP >, true >::operator(), and [Rcpp::sugar::cbind\\_impl::detail::MakeBindableCall](#)< [Rcpp::Vector](#)< LGLSXP >, true >::operator().

### 5.11.1.3 MakeContainerBindable() [3/4]

```
template<>
ContainerBindable<LGLSXP, Rcpp::Vector<LGLSXP> > Rcpp::sugar::cbind_impl::MakeContainerBindable
(
    const Rcpp::Vector< LGLSXP > & x ) [inline]
```

Definition at line 349 of file cbind.h.

### 5.11.1.4 MakeContainerBindable() [4/4]

```
template<int RTYPE>
ContainerBindable<RTYPE, Rcpp::Vector<RTYPE> > Rcpp::sugar::cbind_impl::MakeContainerBindable (
    const Rcpp::Vector< RTYPE > & x ) [inline]
```

Definition at line 337 of file cbind.h.

### 5.11.1.5 MakeScalarBindable()

```
template<typename T >
ScalarBindable<T> Rcpp::sugar::cbind_impl::MakeScalarBindable (
    const T & t ) [inline]
```

Definition at line 355 of file cbind.h.

Referenced by Rcpp::sugar::cbind\_impl::detail::MakeBindableCall< T, false >::operator()().

### 5.11.1.6 operator,()

```
template<int RTYPE, typename E1 , typename E2 >
JoinOp<RTYPE, E1, E2> Rcpp::sugar::cbind_impl::operator, (
    const BindableExpression< RTYPE, E1 > & e1,
    const BindableExpression< RTYPE, E2 > & e2 ) [inline]
```

Definition at line 361 of file cbind.h.

## 5.12 Rcpp::sugar::cbind\_impl::detail Namespace Reference

### Classes

- class [has\\_stored\\_type](#)
- struct [MakeBindableCall](#)
- struct [MakeBindableCall< T, true >](#)
- struct [MakeBindableCall< Rcpp::Matrix< LGLSXP >, true >](#)
- struct [MakeBindableCall< Rcpp::Vector< LGLSXP >, true >](#)
- struct [MakeBindableCall< T, false >](#)
- struct [matrix\\_return](#)
- struct [matrix\\_return< T, true >](#)
- struct [matrix\\_return< Rcpp::Matrix< LGLSXP >, true >](#)
- struct [matrix\\_return< Rcpp::Vector< LGLSXP >, true >](#)
- struct [matrix\\_return< bool, false >](#)
- struct [matrix\\_return< T, false >](#)

### Functions

- [template<typename T >  
Rcpp::traits::enable\\_if< has\\_stored\\_type< T >::value, MakeBindableCall< T, true > >::type MakeBindable \(const T &t\)](#)
- [template<typename T >  
Rcpp::traits::enable\\_if< !has\\_stored\\_type< T >::value, MakeBindableCall< T, false > >::type MakeBindable \(const T &t\)](#)

### 5.12.1 Function Documentation

#### 5.12.1.1 MakeBindable() [1/2]

```
template<typename T >
Rcpp::traits::enable_if< has_stored_type<T>::value, MakeBindableCall<T, true> >::type Rcpp←
::sugar::cbind_impl::detail::MakeBindable (
    const T & t ) [inline]
```

Definition at line 446 of file cbind.h.

#### 5.12.1.2 MakeBindable() [2/2]

```
template<typename T >
Rcpp::traits::enable_if< !has_stored_type<T>::value, MakeBindableCall<T, false> >::type Rcpp←
::sugar::cbind_impl::detail::MakeBindable (
    const T & t ) [inline]
```

Definition at line 455 of file cbind.h.

## 5.13 Rcpp::sugar::detail Namespace Reference

### Classes

- struct [RowSumsReturn](#)
- struct [RowSumsReturn< LGLSXP >](#)
- struct [ColSumsReturn](#)
- struct [RowMeansReturn](#)
- struct [RowMeansReturn< CPLXSP >](#)
- struct [ColMeansReturn](#)

### Functions

- bool [check\\_na](#) (double x)
- bool [check\\_na](#) (int x)
- bool [check\\_na](#) (Rboolean x)
- bool [check\\_na](#) (SEXP x)
- bool [check\\_na](#) (Rcomplex x)
- void [incr](#) (double \*lhs, double rhs)
- void [incr](#) (int \*lhs, int rhs)
- void [incr](#) (Rcomplex \*lhs, const Rcomplex &rhs)
- void [div](#) (double \*lhs, R\_xlen\_t rhs)
- void [div](#) (Rcomplex \*lhs, R\_xlen\_t rhs)
- void [set\\_nan](#) (double \*x)
- void [set\\_nan](#) (Rcomplex \*x)
- bool [isws](#) (const char c)
- const char \* [trim\\_left](#) (const char \*str)
- const char \* [trim\\_right](#) (const char \*str, R\_len\_t sz, std::string \*buff)
- const char \* [trim\\_both](#) (const char \*str, R\_len\_t sz, std::string \*buff)

### 5.13.1 Function Documentation

#### 5.13.1.1 [check\\_na\(\)](#) [1/5]

```
bool Rcpp::sugar::detail::check_na (
    double x ) [inline]
```

Definition at line 30 of file rowSums.h.

Referenced by [Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get\(\)](#), [Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get\(\)](#), [Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get\(\)](#), and [Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get\(\)](#).

**5.13.1.2 check\_na()** [2/5]

```
bool Rcpp::sugar::detail::check_na (
    int x ) [inline]
```

Definition at line 34 of file rowSums.h.

**5.13.1.3 check\_na()** [3/5]

```
bool Rcpp::sugar::detail::check_na (
    Rboolean x ) [inline]
```

Definition at line 38 of file rowSums.h.

**5.13.1.4 check\_na()** [4/5]

```
bool Rcpp::sugar::detail::check_na (
    Rcomplex x ) [inline]
```

Definition at line 46 of file rowSums.h.

**5.13.1.5 check\_na()** [5/5]

```
bool Rcpp::sugar::detail::check_na (
    SEXP x ) [inline]
```

Definition at line 42 of file rowSums.h.

**5.13.1.6 div()** [1/2]

```
void Rcpp::sugar::detail::div (
    double * lhs,
    R_xlen_t rhs ) [inline]
```

Definition at line 65 of file rowSums.h.

Referenced by `Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::get()`, `Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get()`, `Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::get()`, and `Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get()`.

### 5.13.1.7 div() [2/2]

```
void Rcpp::sugar::detail::div (
    Rcomplex * lhs,
    R_xlen_t rhs ) [inline]
```

Definition at line 69 of file rowSums.h.

### 5.13.1.8 incr() [1/3]

```
void Rcpp::sugar::detail::incr (
    double * lhs,
    double rhs ) [inline]
```

Definition at line 51 of file rowSums.h.

Referenced by `Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::get()`, `Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get()`, `Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::get()`, `Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get()`, `Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::get()`, `Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get()`, `Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::get()`, and `Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get()`.

### 5.13.1.9 incr() [2/3]

```
void Rcpp::sugar::detail::incr (
    int * lhs,
    int rhs ) [inline]
```

Definition at line 55 of file rowSums.h.

### 5.13.1.10 incr() [3/3]

```
void Rcpp::sugar::detail::incr (
    Rcomplex * lhs,
    const Rcomplex & rhs ) [inline]
```

Definition at line 59 of file rowSums.h.

### 5.13.1.11 `isws()`

```
bool Rcpp::sugar::detail::isws (
    const char c ) [inline]
```

Definition at line 36 of file trimws.h.

Referenced by `trim_both()`, `trim_left()`, and `trim_right()`.

### 5.13.1.12 `set_nan()` [1/2]

```
void Rcpp::sugar::detail::set_nan (
    double * x ) [inline]
```

Definition at line 75 of file rowSums.h.

Referenced by `Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get()`, and `Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get()`.

### 5.13.1.13 `set_nan()` [2/2]

```
void Rcpp::sugar::detail::set_nan (
    Rcomplex * x ) [inline]
```

Definition at line 79 of file rowSums.h.

### 5.13.1.14 `trim_both()`

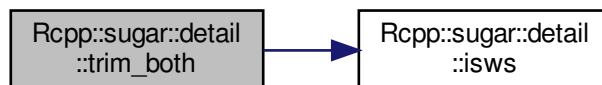
```
const char* Rcpp::sugar::detail::trim_both (
    const char * str,
    R_len_t sz,
    std::string * buff ) [inline]
```

Definition at line 66 of file trimws.h.

References `isws()`.

Referenced by `Rcpp::trimws()`.

Here is the call graph for this function:





### 5.13.1.15 trim\_left()

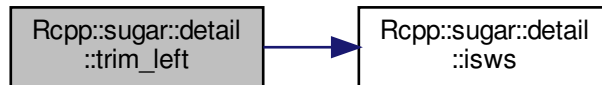
```
const char* Rcpp::sugar::detail::trim_left (
    const char * str ) [inline]
```

Definition at line 40 of file trimws.h.

References `isws()`.

Referenced by `Rcpp::trimws()`.

Here is the call graph for this function:



### 5.13.1.16 trim\_right()

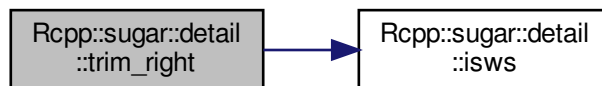
```
const char* Rcpp::sugar::detail::trim_right (
    const char * str,
    R_len_t sz,
    std::string * buff ) [inline]
```

Definition at line 52 of file trimws.h.

References `isws()`.

Referenced by `Rcpp::trimws()`.

Here is the call graph for this function:



## 5.14 Rcpp::sugar::median\_detail Namespace Reference

### Classes

- struct [result](#)
- struct [result< INTSXP >](#)
- struct [result< STRSXP >](#)

### Functions

- template<typename T >  
bool [less](#) (T lhs, T rhs)
- template<> bool [less< Rcomplex >](#) (Rcomplex lhs, Rcomplex rhs)
- double [half](#) (double lhs)
- double [half](#) (int lhs)
- Rcomplex [half](#) (Rcomplex lhs)

### 5.14.1 Function Documentation

#### 5.14.1.1 [half\(\)](#) [1/3]

```
double Rcpp::sugar::median_detail::half (
    double lhs ) [inline]
```

Definition at line 69 of file median.h.

Referenced by [Rcpp::sugar::Median< RTYPE, NA, T, NA\\_RM >::operator result\\_type\(\)](#), [Rcpp::sugar::Median< RTYPE, NA, T, true >::operator result\\_type\(\)](#), and [Rcpp::sugar::Median< RTYPE, false, T, NA\\_RM >::operator result\\_type\(\)](#).

#### 5.14.1.2 [half\(\)](#) [2/3]

```
double Rcpp::sugar::median_detail::half (
    int lhs ) [inline]
```

Definition at line 73 of file median.h.

### 5.14.1.3 half() [3/3]

```
Rcomplex Rcpp::sugar::median_detail::half (  
    Rcomplex lhs ) [inline]
```

Definition at line 77 of file median.h.

### 5.14.1.4 less()

```
template<typename T >  
bool Rcpp::sugar::median_detail::less (  
    T lhs,  
    T rhs ) [inline]
```

Definition at line 55 of file median.h.

### 5.14.1.5 less< Rcomplex >()

```
template<>  
bool Rcpp::sugar::median_detail::less< Rcomplex > (  
    Rcomplex lhs,  
    Rcomplex rhs ) [inline]
```

Definition at line 60 of file median.h.

## 5.15 Rcpp::traits Namespace Reference

traits used to dispatch wrap

### Classes

- struct [wrap\\_type\\_traits< Rcpp::Date >](#)
- struct [r\\_type\\_traits< Rcpp::Date >](#)
- struct [r\\_type\\_traits< std::pair< const std::string, Rcpp::Date > >](#)
- struct [r\\_sexptype\\_traits< Rcpp::Date >](#)
- struct [wrap\\_type\\_traits< Rcpp::Datetime >](#)
- struct [r\\_type\\_traits< Rcpp::Datetime >](#)
- struct [r\\_type\\_traits< std::pair< const std::string, Rcpp::Datetime > >](#)
- struct [r\\_sexptype\\_traits< Rcpp::Datetime >](#)
- class [ContainerExporter](#)
- struct [container\\_exporter< Container, int >](#)
- struct [container\\_exporter< Container, double >](#)

- struct [Extractor](#)
- struct [input\\_parameter](#)
- struct [input\\_parameter< T >](#)
- struct [input\\_parameter< const T >](#)
- struct [input\\_parameter< const T & >](#)
- struct [r\\_type\\_traits< Rcpp::String >](#)
- struct [r\\_sexptype\\_traits< Rcpp::String >](#)
- struct [char\\_type](#)
- struct [char\\_type< const wchar\\_t \\* >](#)
- struct [char\\_type< const char \\* >](#)
- struct [enable\\_if](#)
- struct [enable\\_if< true, T >](#)
- struct [expands\\_to\\_logical\\_\\_impl](#)
- struct [expands\\_to\\_logical\\_\\_impl< LGLSXP >](#)
- class [\\_has\\_rtype\\_helper](#)
- struct [expands\\_to\\_logical](#)
- struct [\\_\\_sfnaf\\_types](#)
- class [\\_has\\_iterator\\_helper](#)
- class [\\_is\\_importer\\_helper](#)
- class [\\_is\\_generator\\_helper](#)
- class [\\_is\\_exporter\\_helper](#)
- struct [has\\_iterator](#)
- struct [is\\_importer](#)
- struct [is\\_exporter](#)
- struct [is\\_generator](#)
- struct [has\\_na](#)
- struct [has\\_na< INTSXP >](#)
- struct [has\\_na< REALSXP >](#)
- struct [has\\_na< CPLXSXP >](#)
- struct [has\\_na< STRSXP >](#)
- struct [has\\_na< LGLSXP >](#)
- struct [if\\_](#)
- struct [if\\_< false, LHS, RHS >](#)
- struct [init\\_type](#)
- struct [init\\_type< STRSXP >](#)
- struct [init\\_type< LGLSXP >](#)
- struct [integral\\_constant](#)
- struct [both](#)
- struct [is\\_arithmetic](#)
- struct [is\\_arithmetic< short >](#)
- struct [is\\_arithmetic< const short >](#)
- struct [is\\_arithmetic< unsigned short >](#)
- struct [is\\_arithmetic< const unsigned short >](#)
- struct [is\\_arithmetic< int >](#)
- struct [is\\_arithmetic< const int >](#)
- struct [is\\_arithmetic< unsigned int >](#)
- struct [is\\_arithmetic< const unsigned int >](#)
- struct [is\\_arithmetic< long >](#)
- struct [is\\_arithmetic< const long >](#)
- struct [is\\_arithmetic< unsigned long >](#)
- struct [is\\_arithmetic< const unsigned long >](#)

- struct [is\\_arithmetic< float >](#)
- struct [is\\_arithmetic< const float >](#)
- struct [is\\_arithmetic< double >](#)
- struct [is\\_arithmetic< const double >](#)
- struct [is\\_arithmetic< long double >](#)
- struct [is\\_arithmetic< const long double >](#)
- struct [is\\_bool](#)
- struct [is\\_bool< bool >](#)
- struct [is\\_bool< const bool >](#)
- struct [is\\_bool< volatile bool >](#)
- struct [is\\_const](#)
  - type properties [4.5.3].*
- struct [is\\_const< \\_Tp const >](#)
- class [is\\_convertible](#)
- class [\\_is\\_eigen\\_helper](#)
- struct [is\\_eigen\\_base](#)
- struct [is\\_module\\_object](#)
- struct [is\\_pointer](#)
- struct [is\\_pointer< T \\* >](#)
- struct [is\\_primitive](#)
- struct [is\\_reference](#)
- struct [is\\_reference< \\_Tp & >](#)
- class [\\_is\\_sugar\\_expression\\_helper](#)
- struct [is\\_sugar\\_expression](#)
- struct [is\\_trivial](#)
- struct [is\\_trivial< VECSXP >](#)
- struct [is\\_trivial< EXPRSXP >](#)
- struct [is\\_wide\\_string](#)
- struct [is\\_wide\\_string< const wchar\\_t \\* >](#)
- struct [is\\_wide\\_string< const char \\* >](#)
- struct [is\\_wide\\_string< wchar\\_t >](#)
- struct [is\\_wide\\_string< char >](#)
- class [\\_has\\_matrix\\_interface\\_helper](#)
- struct [matrix\\_interface](#)
- struct [normal\\_wrap\\_tag](#)
- struct [void\\_wrap\\_tag](#)
- struct [pointer\\_wrap\\_tag](#)
- struct [module\\_wrap\\_traits](#)
- struct [module\\_wrap\\_traits< void >](#)
- struct [module\\_wrap\\_traits< T \\* >](#)
- struct [needs\\_protection](#)
- struct [needs\\_protection< SEXP >](#)
- class [named\\_object](#)
- class [named\\_object< SEXP >](#)
- struct [is\\_named](#)
- struct [is\\_named< named\\_object< T > >](#)
- struct [is\\_named< Rcpp::Argument >](#)
- struct [num2type](#)
- struct [allowed\\_matrix\\_type](#)
- struct [allowed\\_matrix\\_type< true >](#)

- class [one\\_type](#)
- class [zero\\_type](#)
- struct [r\\_sexptype\\_traits](#)
- struct [r\\_sexptype\\_traits< int >](#)
- struct [r\\_sexptype\\_traits< const int >](#)
- struct [r\\_sexptype\\_traits< double >](#)
- struct [r\\_sexptype\\_traits< const double >](#)
- struct [r\\_sexptype\\_traits< bool >](#)
- struct [r\\_sexptype\\_traits< std::string >](#)
- struct [r\\_sexptype\\_traits< Rcomplex >](#)
- struct [r\\_sexptype\\_traits< Rbyte >](#)
- struct [r\\_sexptype\\_traits< unsigned int >](#)
- struct [r\\_sexptype\\_traits< float >](#)
- struct [r\\_sexptype\\_traits< long >](#)
- struct [r\\_sexptype\\_traits< unsigned long >](#)
- struct [r\\_sexptype\\_traits< long double >](#)
- struct [r\\_sexptype\\_traits< short >](#)
- struct [r\\_sexptype\\_traits< unsigned short >](#)
- struct [r\\_sexptype\\_traits< std::complex< double > >](#)
- struct [r\\_sexptype\\_traits< std::complex< float > >](#)
- struct [r\\_sexptype\\_needs cast](#)
- struct [r\\_sexptype\\_needs cast< int >](#)
- struct [r\\_sexptype\\_needs cast< double >](#)
- struct [r\\_sexptype\\_needs cast< Rcomplex >](#)
- struct [r\\_sexptype\\_needs cast< Rbyte >](#)
- struct [r\\_type\\_primitive\\_tag](#)
- struct [r\\_type\\_string\\_tag](#)
- struct [r\\_type\\_generic\\_tag](#)
- struct [r\\_type\\_pairstring\\_primitive\\_tag](#)
- struct [r\\_type\\_pairstring\\_string\\_tag](#)
- struct [r\\_type\\_pairstring\\_generic\\_tag](#)
- struct [r\\_type\\_pair\\_tag](#)
- struct [r\\_type\\_module\\_object\\_pointer\\_tag](#)
- struct [r\\_type\\_module\\_object\\_const\\_pointer\\_tag](#)
- struct [r\\_type\\_module\\_object\\_tag](#)
- struct [r\\_type\\_module\\_object\\_reference\\_tag](#)
- struct [r\\_type\\_module\\_object\\_const\\_reference\\_tag](#)
- struct [r\\_type\\_enum\\_tag](#)
- struct [r\\_type\\_RcppString\\_tag](#)
- struct [r\\_type\\_traits](#)
- struct [r\\_type\\_traits< Rcpp::object< T > >](#)
- struct [r\\_type\\_traits< std::pair< const KEY, VALUE > >](#)
- struct [r\\_type\\_traits< std::pair< const std::string, T > >](#)
- struct [r\\_type\\_traits< std::pair< const std::string, int > >](#)
- struct [r\\_type\\_traits< std::pair< const std::string, const int > >](#)
- struct [r\\_type\\_traits< std::pair< const std::string, double > >](#)
- struct [r\\_type\\_traits< std::pair< const std::string, Rbyte > >](#)
- struct [r\\_type\\_traits< std::pair< const std::string, Rcomplex > >](#)
- struct [r\\_type\\_traits< std::pair< const std::string, bool > >](#)
- struct [r\\_type\\_traits< std::pair< const std::string, std::string > >](#)
- struct [r\\_type\\_traits< std::pair< const std::string, std::wstring > >](#)

- struct `r_type_traits< std::pair< const std::string, char > >`
- struct `r_type_traits< std::pair< const std::string, wchar_t > >`
- struct `r_type_traits< std::pair< const std::string, unsigned int > >`
- struct `r_type_traits< std::pair< const std::string, float > >`
- struct `r_type_traits< int >`
- struct `r_type_traits< const int >`
- struct `r_type_traits< double >`
- struct `r_type_traits< const double >`
- struct `r_type_traits< Rbyte >`
- struct `r_type_traits< Rcomplex >`
- struct `r_type_traits< bool >`
- struct `r_type_traits< std::string >`
- struct `r_type_traits< std::wstring >`
- struct `r_type_traits< char >`
- struct `r_type_traits< wchar_t >`
- struct `r_type_traits< unsigned int >`
- struct `r_type_traits< float >`
- struct `r_type_traits< const char * >`
- struct `r_type_traits< const wchar_t * >`
- struct `r_type_traits< long >`
- struct `r_type_traits< std::pair< const std::string, long > >`
- struct `r_type_traits< unsigned long >`
- struct `r_type_traits< std::pair< const std::string, unsigned long > >`
- struct `r_type_traits< long double >`
- struct `r_type_traits< std::pair< const std::string, long double > >`
- struct `r_type_traits< short >`
- struct `r_type_traits< std::pair< const std::string, short > >`
- struct `r_type_traits< unsigned short >`
- struct `r_type_traits< std::pair< const std::string, unsigned short > >`
- struct `r_type_traits< std::complex< double > >`
- struct `r_type_traits< std::pair< const std::string, std::complex< double > > >`
- struct `r_type_traits< std::complex< float > >`
- struct `r_type_traits< std::pair< const std::string, std::complex< float > > >`
- struct `remove_const`
  - *const-volatile modifications [4.7.1].*
  - struct `remove_const< _Tp const >`
  - struct `remove_const_and_reference`
  - struct `remove_reference`
    - *reference modifications [4.7.2].*
    - struct `remove_reference< _Tp & >`
- struct `result_of`
- struct `result_of< RESULT_TYPE(*) (INPUT_TYPE) >`
- struct `result_of< RESULT_TYPE(*) (U1, U2) >`
- struct `result_of< RESULT_TYPE(*) (U1, U2, U3) >`
- struct `same_type`
- struct `same_type< T, T >`
- struct `storage_type`
- struct `storage_type< INTSXP >`
- struct `storage_type< REALSXP >`
- struct `storage_type< CPLXSXP >`

- struct [storage\\_type< RAWSXP >](#)
- struct [storage\\_type< LGLSXP >](#)
- struct [identity](#)
- struct [int2type](#)
- struct [un\\_pointer](#)
- struct [un\\_pointer< T \\* >](#)
- struct [un\\_pointer< object< T > >](#)
- struct [wrap\\_type\\_primitive\\_tag](#)
- struct [wrap\\_type\\_unknown\\_tag](#)
- struct [wrap\\_type\\_module\\_object\\_pointer\\_tag](#)
- struct [wrap\\_type\\_module\\_object\\_tag](#)
- struct [wrap\\_type\\_enum\\_tag](#)
- struct [wrap\\_type\\_char\\_array](#)
- struct [wrap\\_type\\_traits](#)
- struct [wrap\\_type\\_traits< char\[N\]>](#)
- struct [wrap\\_type\\_traits< const char\[N\]>](#)
- struct [wrap\\_type\\_traits< int >](#)
- struct [wrap\\_type\\_traits< const int >](#)
- struct [wrap\\_type\\_traits< double >](#)
- struct [wrap\\_type\\_traits< Rbyte >](#)
- struct [wrap\\_type\\_traits< Rcomplex >](#)
- struct [wrap\\_type\\_traits< unsigned int >](#)
- struct [wrap\\_type\\_traits< bool >](#)
- struct [wrap\\_type\\_traits< std::string >](#)
- struct [wrap\\_type\\_traits< std::wstring >](#)
- struct [wrap\\_type\\_traits< Rcpp::String >](#)
- struct [wrap\\_type\\_traits< char >](#)
- struct [wrap\\_type\\_traits< wchar\\_t >](#)
- struct [wrap\\_type\\_traits< float >](#)
- struct [wrap\\_type\\_traits< std::complex< float > >](#)
- struct [wrap\\_type\\_traits< std::complex< double > >](#)
- struct [wrap\\_type\\_traits< long >](#)
- struct [wrap\\_type\\_traits< unsigned long >](#)
- struct [wrap\\_type\\_traits< long double >](#)
- struct [wrap\\_type\\_traits< short >](#)
- struct [wrap\\_type\\_traits< unsigned short >](#)
- struct [wrap\\_type\\_traits< Rcpp::object< T > >](#)
- struct [r\\_vector\\_cache\\_type](#)
- class [r\\_vector\\_cache](#)
- struct [r\\_vector\\_name\\_proxy](#)
- struct [r\\_vector\\_proxy](#)
- struct [r\\_vector\\_const\\_proxy](#)
- struct [r\\_vector\\_iterator](#)
- struct [r\\_vector\\_const\\_iterator](#)
- struct [r\\_vector\\_element\\_converter](#)
- struct [r\\_vector\\_element\\_converter< STRSXP >](#)
- struct [r\\_vector\\_element\\_converter< VECSXP >](#)
- struct [r\\_vector\\_element\\_converter< EXPRSXP >](#)
- struct [r\\_vector\\_name\\_proxy< STRSXP, StoragePolicy >](#)
- struct [r\\_vector\\_name\\_proxy< VECSXP, StoragePolicy >](#)
- struct [r\\_vector\\_name\\_proxy< EXPRSXP, StoragePolicy >](#)



- struct [r\\_vector\\_proxy](#)< STRSXP, StoragePolicy >
- struct [r\\_vector\\_proxy](#)< EXPRSXP, StoragePolicy >
- struct [r\\_vector\\_proxy](#)< VECSXP, StoragePolicy >
- struct [r\\_vector\\_const\\_proxy](#)< STRSXP, StoragePolicy >
- struct [r\\_vector\\_const\\_proxy](#)< VECSXP, StoragePolicy >
- struct [r\\_vector\\_const\\_proxy](#)< EXPRSXP, StoragePolicy >
- struct [proxy\\_based\\_iterator](#)
- struct [r\\_vector\\_iterator](#)< VECSXP, StoragePolicy >
- struct [r\\_vector\\_iterator](#)< EXPRSXP, StoragePolicy >
- struct [r\\_vector\\_iterator](#)< STRSXP, StoragePolicy >
- struct [proxy\\_based\\_const\\_iterator](#)
- struct [r\\_vector\\_const\\_iterator](#)< VECSXP, StoragePolicy >
- struct [r\\_vector\\_const\\_iterator](#)< EXPRSXP, StoragePolicy >
- struct [r\\_vector\\_const\\_iterator](#)< STRSXP, StoragePolicy >
- class [proxy\\_cache](#)
- struct [r\\_vector\\_cache\\_type](#)< VECSXP, StoragePolicy >
- struct [r\\_vector\\_cache\\_type](#)< EXPRSXP, StoragePolicy >
- struct [r\\_vector\\_cache\\_type](#)< STRSXP, StoragePolicy >

## Typedefs

- typedef [integral\\_constant](#)< bool, true > [true\\_type](#)
- typedef [integral\\_constant](#)< bool, false > [false\\_type](#)

## Functions

- template<int RTYPE>  
[storage\\_type](#)< RTYPE >::type [get\\_na](#) ()
- template<> int [get\\_na](#)< INTSXP > ()
- template<> int [get\\_na](#)< LGLSXP > ()
- template<> double [get\\_na](#)< REALSXP > ()
- template<> Rcomplex [get\\_na](#)< CPLXSXP > ()
- template<> SEXP [get\\_na](#)< STRSXP > ()
- template<> SEXP [get\\_na](#)< VECSXP > ()
- template<int RTYPE>  
bool [is\\_finite](#) (typename [storage\\_type](#)< RTYPE >::type)
- template<> bool [is\\_finite](#)< INTSXP > (int x)
- template<> bool [is\\_finite](#)< REALSXP > (double x)
- template<> bool [is\\_finite](#)< CPLXSXP > (Rcomplex x)
- template<> bool [is\\_finite](#)< STRSXP > (SEXP)
- template<> bool [is\\_finite](#)< LGLSXP > (int x)
- template<int RTYPE>  
bool [is\\_infinite](#) (typename [storage\\_type](#)< RTYPE >::type)
- template<> bool [is\\_infinite](#)< REALSXP > (double x)
- template<> bool [is\\_infinite](#)< CPLXSXP > (Rcomplex x)
- template<int RTYPE>  
bool [is\\_na](#) (typename [storage\\_type](#)< RTYPE >::type)
- template<> bool [is\\_na](#)< INTSXP > (int x)
- template<> bool [is\\_na](#)< REALSXP > (double x)

- `template<> bool is_na< CPLXSXP >` (Rcomplex x)
- `template<> bool is_na< STRSXP >` (SEXP x)
- `template<> bool is_na< LGLSXP >` (int x)
- `template<int RTYPE>`  
`bool is_nan` (typename `storage_type< RTYPE >::type`)
- `template<> bool is_nan< REALSXP >` (double x)
- `template<> bool is_nan< CPLXSXP >` (Rcomplex x)

### 5.15.1 Detailed Description

traits used to dispatch wrap

### 5.15.2 Typedef Documentation

#### 5.15.2.1 false\_type

```
typedef integral\_constant<bool, false> Rcpp::traits::false\_type
```

Definition at line 35 of file `integral_constant.h`.

#### 5.15.2.2 true\_type

```
typedef integral\_constant<bool, true> Rcpp::traits::true\_type
```

Definition at line 34 of file `integral_constant.h`.

### 5.15.3 Function Documentation

#### 5.15.3.1 get\_na()

```
template<int RTYPE>
storage\_type<RTYPE>::type Rcpp::traits::get\_na ( )
```

### 5.15.3.2 `get_na< CPLXSXP >()`

```
template<>
Rcomplex Rcpp::traits::get_na< CPLXSXP > ( ) [inline]
```

Definition at line 42 of file `get_na.h`.

### 5.15.3.3 `get_na< INTSXP >()`

```
template<>
int Rcpp::traits::get_na< INTSXP > ( ) [inline]
```

Definition at line 33 of file `get_na.h`.

### 5.15.3.4 `get_na< LGLSXP >()`

```
template<>
int Rcpp::traits::get_na< LGLSXP > ( ) [inline]
```

Definition at line 36 of file `get_na.h`.

### 5.15.3.5 `get_na< REALSXP >()`

```
template<>
double Rcpp::traits::get_na< REALSXP > ( ) [inline]
```

Definition at line 39 of file `get_na.h`.

Referenced by `Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::operator[]()`.

### 5.15.3.6 `get_na< STRSXP >()`

```
template<>
SEXP Rcpp::traits::get_na< STRSXP > ( ) [inline]
```

Definition at line 50 of file `get_na.h`.

Referenced by `Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::operator result_type()`, `Rcpp::sugar::Median< STRSXP, NA, T, true >::operator result_type()`, and `Rcpp::sugar::Median< STRSXP, false, T, true >::operator result_type()`.

### 5.15.3.7 `get_na< VECSXP >()`

```
template<>
SEXP Rcpp::traits::get_na< VECSXP > ( ) [inline]
```

Definition at line 54 of file `get_na.h`.

### 5.15.3.8 `is_finite()`

```
template<int RTYPE>
bool Rcpp::traits::is_finite (
    typename storage_type< RTYPE >::type )
```

### 5.15.3.9 `is_finite< CPLXSXP >()`

```
template<>
bool Rcpp::traits::is_finite< CPLXSXP > (
    Rcomplex x ) [inline]
```

Definition at line 44 of file `is_finite.h`.

### 5.15.3.10 `is_finite< INTSXP >()`

```
template<>
bool Rcpp::traits::is_finite< INTSXP > (
    int x ) [inline]
```

Definition at line 34 of file `is_finite.h`.

### 5.15.3.11 `is_finite< LGLSXP >()`

```
template<>
bool Rcpp::traits::is_finite< LGLSXP > (
    int x ) [inline]
```

Definition at line 54 of file `is_finite.h`.

### 5.15.3.12 is\_finite< REALSXP >()

```
template<>
bool Rcpp::traits::is_finite< REALSXP > (
    double x ) [inline]
```

Definition at line 39 of file is\_finite.h.

### 5.15.3.13 is\_finite< STRSXP >()

```
template<>
bool Rcpp::traits::is_finite< STRSXP > (
    SEXP ) [inline]
```

Definition at line 49 of file is\_finite.h.

### 5.15.3.14 is\_infinite()

```
template<int RTYPE>
bool Rcpp::traits::is_infinite (
    typename storage_type< RTYPE >::type )
```

Definition at line 30 of file is\_infinite.h.

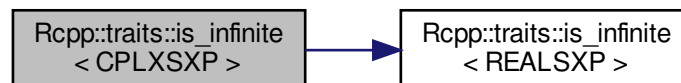
### 5.15.3.15 is\_infinite< CPLXSXP >()

```
template<>
bool Rcpp::traits::is_infinite< CPLXSXP > (
    Rcomplex x ) [inline]
```

Definition at line 40 of file is\_infinite.h.

References is\_infinite< REALSXP >().

Here is the call graph for this function:



**5.15.3.16 is\_infinite< REALSXP >()**

```
template<>
bool Rcpp::traits::is_infinite< REALSXP > (
    double x ) [inline]
```

Definition at line 35 of file is\_infinite.h.

Referenced by is\_infinite< CPLXSXP >().

**5.15.3.17 is\_na()**

```
template<int RTYPE>
bool Rcpp::traits::is_na (
    typename storage_type< RTYPE >::type )
```

Definition at line 32 of file is\_na.h.

**5.15.3.18 is\_na< CPLXSXP >()**

```
template<>
bool Rcpp::traits::is_na< CPLXSXP > (
    Rcomplex x ) [inline]
```

Definition at line 47 of file is\_na.h.

Referenced by Rcpp::sugar::not\_< CPLXSXP, NA >::apply(), Rcpp::sugar::unary\_minus< CPLXSXP, NA >::apply(), and Rcpp::sugar::SugarComplex< NA, RESULT\_TYPE, T, FunPtr >::operator[]().

**5.15.3.19 is\_na< INTSXP >()**

```
template<>
bool Rcpp::traits::is_na< INTSXP > (
    int x ) [inline]
```

Definition at line 37 of file is\_na.h.

Referenced by Rcpp::sugar::SugarMath\_1< NA, RESULT\_TYPE, int, T1, FunPtr >::operator[]().

**5.15.3.20 is\_na< LGLSXP >()**

```
template<>
bool Rcpp::traits::is_na< LGLSXP > (
    int x ) [inline]
```

Definition at line 57 of file is\_na.h.

Referenced by Rcpp::sugar::All< NA, T >::apply(), Rcpp::sugar::Any< NA, T >::apply(), Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::apply(), Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::apply(), Rcpp::sugar::negate< NA >::apply(), Rcpp::sugar::SingleLogicalResult< NA, T >::is\_na(), Rcpp::sugar::IfElse< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, COND\_NA, COND\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T >::operator[](), and Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::operator[]().

**5.15.3.21 is\_na< REALSXP >()**

```
template<>
bool Rcpp::traits::is_na< REALSXP > (
    double x ) [inline]
```

Definition at line 42 of file is\_na.h.

Referenced by Rcpp::sugar::not\_< REALSXP, NA >::apply(), Rcpp::Date::is\_na(), Rcpp::Datetime::is\_na(), Rcpp::sugar::pmax\_op\_Vector\_Primitive< REALSXP, true >::operator(), Rcpp::sugar::pmin\_op\_Vector\_Primitive< REALSXP, true >::operator(), Rcpp::sugar::pmax\_op< REALSXP, true, true >::operator(), Rcpp::sugar::pmin\_op< REALSXP, true, true >::operator(), and Rcpp::sugar::clamp\_operator< REALSXP, true >::operator().

**5.15.3.22 is\_na< STRSXP >()**

```
template<>
bool Rcpp::traits::is_na< STRSXP > (
    SEXP x ) [inline]
```

Definition at line 52 of file is\_na.h.

Referenced by Rcpp::trimws().

### 5.15.3.23 `is_nan()`

```
template<int RTYPE>
bool Rcpp::traits::is_nan (
    typename storage_type< RTYPE >::type )
```

Definition at line 30 of file `is_nan.h`.

### 5.15.3.24 `is_nan< CPLXSXP >()`

```
template<>
bool Rcpp::traits::is_nan< CPLXSXP > (
    Rcomplex x ) [inline]
```

Definition at line 40 of file `is_nan.h`.

### 5.15.3.25 `is_nan< REALSXP >()`

```
template<>
bool Rcpp::traits::is_nan< REALSXP > (
    double x ) [inline]
```

Definition at line 35 of file `is_nan.h`.

## 5.16 `std` Namespace Reference

## 5.17 `tinyformat` Namespace Reference

### Namespaces

- [detail](#)

### Classes

- class [FormatList](#)

### Typedefs

- typedef const [FormatList](#) & [FormatListRef](#)  
*Reference to type-opaque format list for passing to `vformat()`*



## Functions

- `template<typename T >`  
void `formatValue` (`std::ostream &out`, `const char *`, `const char *fmtEnd`, `int ntrunc`, `const T &value`)
- `detail::FormatListN< 0 >` `makeFormatList` ()
- void `vformat` (`std::ostream &out`, `const char *fmt`, `FormatListRef` list)
- void `format` (`std::ostream &out`, `const char *fmt`)
- `std::string` `format` (`const char *fmt`)
- void `printf` (`const char *fmt`)
- void `printfln` (`const char *fmt`)

### 5.17.1 Typedef Documentation

#### 5.17.1.1 FormatListRef

```
typedef const FormatList& tinyformat::FormatListRef
```

Reference to type-opaque format list for passing to `vformat()`

Definition at line 867 of file `tinyformat.h`.

### 5.17.2 Function Documentation

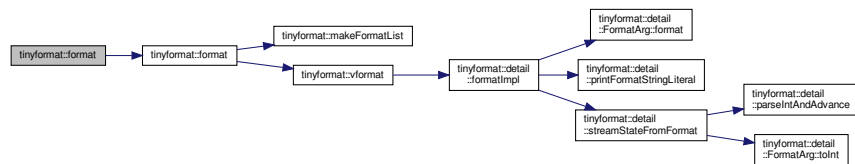
#### 5.17.2.1 format() [1/2]

```
std::string tinyformat::format (
    const char * fmt ) [inline]
```

Definition at line 1001 of file `tinyformat.h`.

References `format()`.

Here is the call graph for this function:



### 5.17.2.2 format() [2/2]

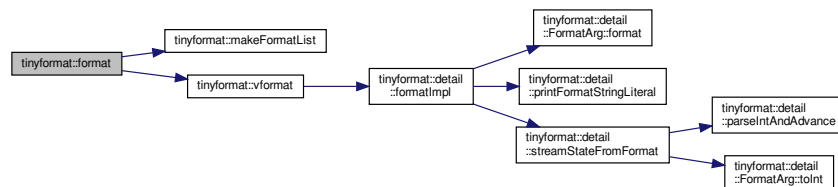
```
void tinyformat::format (
    std::ostream & out,
    const char * fmt ) [inline]
```

Definition at line 996 of file tinyformat.h.

References `makeFormatList()`, and `vformat()`.

Referenced by `format()`, `printf()`, `printfln()`, `Rcpp::sprintf()`, `Rcpp::stop()`, and `Rcpp::warning()`.

Here is the call graph for this function:



### 5.17.2.3 formatValue()

```
template<typename T >
void tinyformat::formatValue (
    std::ostream & out,
    const char * ,
    const char * fmtEnd,
    int ntrunc,
    const T & value ) [inline]
```

Format a value into a stream, delegating to `operator<<` by default.

Users may override this for their own types. When this function is called, the stream flags will have been modified according to the format string. The format specification is provided in the range `[fmtBegin, fmtEnd)`. For truncating conversions, `ntrunc` is set to the desired maximum number of characters, for example `"%.7s"` calls `formatValue` with `ntrunc = 7`.

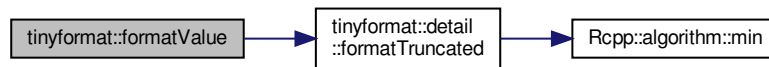
By default, `formatValue()` uses the usual stream insertion operator `operator<<` to format the type `T`, with special cases for the `c` and `p` conversions.

Definition at line 312 of file tinyformat.h.

References `tinyformat::detail::formatTruncated()`.

Referenced by `tinyformat::detail::FormatArg::formatImpl()`.

Here is the call graph for this function:



#### 5.17.2.4 makeFormatList()

```
detail::FormatListN<O> tinyformat::makeFormatList ( ) [inline]
```

Definition at line 935 of file tinyformat.h.

Referenced by format().

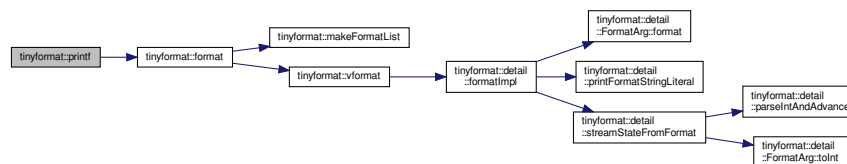
#### 5.17.2.5 printf()

```
void tinyformat::printf (
    const char * fmt ) [inline]
```

Definition at line 1008 of file tinyformat.h.

References format().

Here is the call graph for this function:



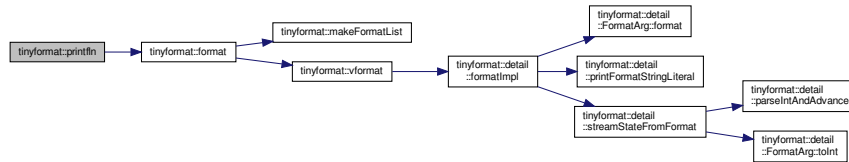
### 5.17.2.6 printfn()

```
void tinyformat::printfn (
    const char * fmt ) [inline]
```

Definition at line 1013 of file tinyformat.h.

References [format\(\)](#).

Here is the call graph for this function:



### 5.17.2.7 vformat()

```
void tinyformat::vformat (
    std::ostream & out,
    const char * fmt,
    FormatListRef list ) [inline]
```

Format list of arguments to the stream according to the given format string.

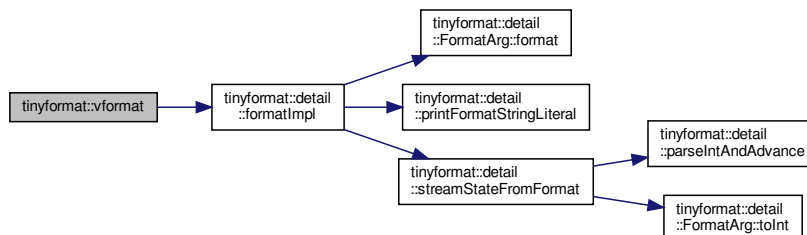
The name [vformat\(\)](#) is chosen for the semantic similarity to `vprintf()`: the list of format arguments is held in a single function argument.

Definition at line 954 of file tinyformat.h.

References [tinyformat::detail::formatImpl\(\)](#), [tinyformat::FormatList::m\\_formatters](#), and [tinyformat::FormatList::m\\_N](#).

Referenced by [format\(\)](#).

Here is the call graph for this function:



## 5.18 tinyformat::detail Namespace Reference

### Classes

- struct [is\\_convertible](#)
- struct [is\\_wchar](#)
- struct [is\\_wchar< wchar\\_t \\* >](#)
- struct [is\\_wchar< const wchar\\_t \\* >](#)
- struct [is\\_wchar< const wchar\\_t\[n\]>](#)
- struct [is\\_wchar< wchar\\_t\[n\]>](#)
- struct [formatValueAsType](#)
- struct [formatValueAsType< T, fmtT, true >](#)
- struct [convertToInt](#)
- struct [convertToInt< T, true >](#)
- class [FormatArg](#)
- class [FormatListN](#)
- class [FormatListN< 0 >](#)

### Functions

- template<typename T >  
void [formatTruncated](#) (std::ostream &out, const T &value, int ntrunc)
- int [parseIntAndAdvance](#) (const char \*&c)
- const char \* [printFormatStringLiteral](#) (std::ostream &out, const char \*fmt)
- const char \* [streamStateFromFormat](#) (std::ostream &out, bool &spacePadPositive, int &ntrunc, const char \*fmt↔  
Start, const [detail::FormatArg](#) \*formatters, int &argIndex, int numFormatters)
- void [formatImpl](#) (std::ostream &out, const char \*fmt, const [detail::FormatArg](#) \*formatters, int numFormatters)

### 5.18.1 Function Documentation

#### 5.18.1.1 formatImpl()

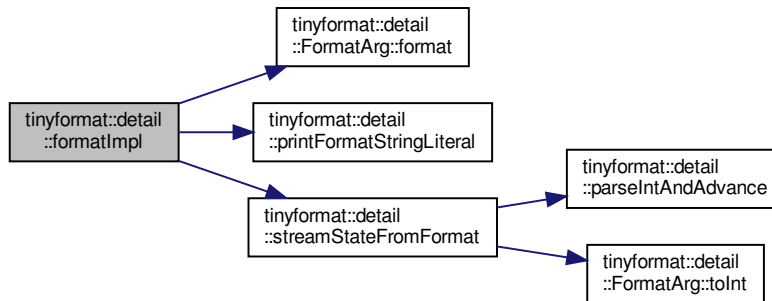
```
void tinyformat::detail::formatImpl (
    std::ostream & out,
    const char * fmt,
    const detail::FormatArg * formatters,
    int numFormatters ) [inline]
```

Definition at line 785 of file tinyformat.h.

References [tinyformat::detail::FormatArg::format\(\)](#), [printFormatStringLiteral\(\)](#), [streamStateFromFormat\(\)](#), and [TINYFORMAT\\_ERROR](#).

Referenced by [tinyformat::vformat\(\)](#).

Here is the call graph for this function:



### 5.18.1.2 formatTruncated()

```

template<typename T >
void tinyformat::detail::formatTruncated (
    std::ostream & out,
    const T & value,
    int ntrunc ) [inline]
  
```

Definition at line 271 of file `tinyformat.h`.

References `Rcpp::algorithm::min()`.

Referenced by `tinyformat::formatValue()`.

Here is the call graph for this function:



### 5.18.1.3 parseIntAndAdvance()

```
int tinyformat::detail::parseIntAndAdvance (
    const char *& c ) [inline]
```

Definition at line 546 of file tinyformat.h.

Referenced by streamStateFromFormat().

### 5.18.1.4 printFormatStringLiteral()

```
const char* tinyformat::detail::printFormatStringLiteral (
    std::ostream & out,
    const char * fmt ) [inline]
```

Definition at line 560 of file tinyformat.h.

Referenced by formatImpl().

### 5.18.1.5 streamStateFromFormat()

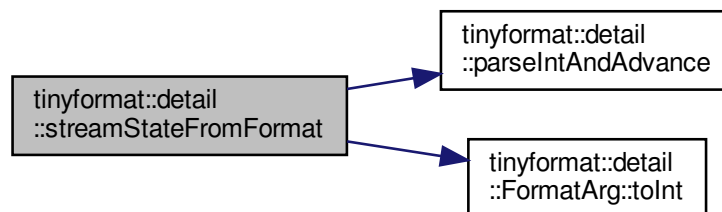
```
const char* tinyformat::detail::streamStateFromFormat (
    std::ostream & out,
    bool & spacePadPositive,
    int & ntrunc,
    const char * fmtStart,
    const detail::FormatArg * formatters,
    int & argIndex,
    int numFormatters ) [inline]
```

Definition at line 594 of file tinyformat.h.

References parseIntAndAdvance(), TINYFORMAT\_ERROR, and tinyformat::detail::FormatArg::toInt().

Referenced by formatImpl().

Here is the call graph for this function:







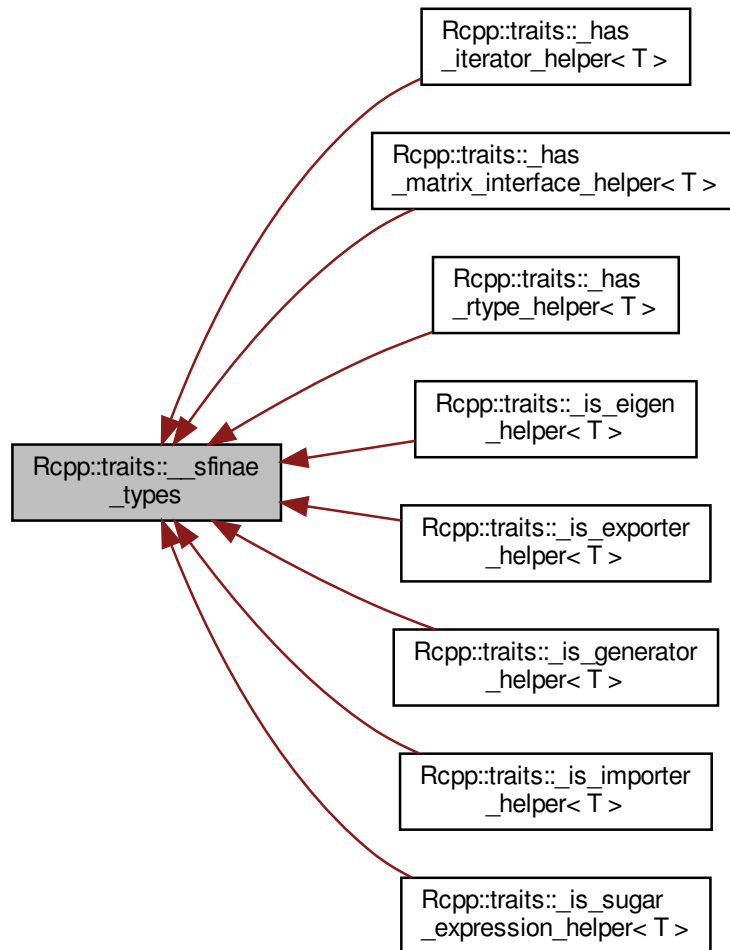
## Chapter 6

# Class Documentation

### 6.1 Rcpp::traits::\_\_sfinae\_types Struct Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::\_\_sfniae\_types:



## Classes

- struct [\\_\\_two](#)

## Public Types

- typedef char [\\_\\_one](#)

### 6.1.1 Detailed Description

Definition at line 34 of file `has_iterator.h`.

## 6.1.2 Member Typedef Documentation

### 6.1.2.1 \_\_one

```
typedef char Rcpp::traits::__sfinae_types::__one
```

Definition at line 35 of file `has_iterator.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/has_iterator.h`

## 6.2 Rcpp::traits::\_\_sfinae\_types::\_\_two Struct Reference

```
#include <has_iterator.h>
```

### Public Attributes

- `char __arr [2]`

### 6.2.1 Detailed Description

Definition at line 36 of file `has_iterator.h`.

### 6.2.2 Member Data Documentation

#### 6.2.2.1 \_\_arr

```
char Rcpp::traits::__sfinae_types::__two::__arr[2]
```

Definition at line 36 of file `has_iterator.h`.

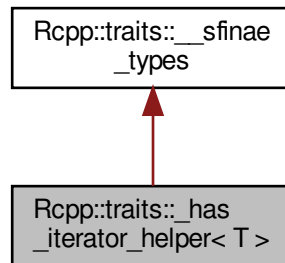
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/has_iterator.h`

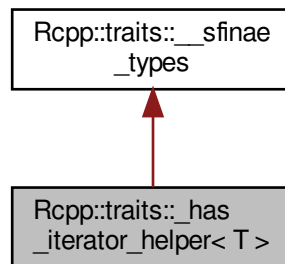
### 6.3 Rcpp::traits::\_has\_iterator\_helper< T > Class Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::\_has\_iterator\_helper< T >:



Collaboration diagram for Rcpp::traits::\_has\_iterator\_helper< T >:



#### Classes

- struct [\\_Wrap\\_type](#)

#### Static Public Attributes

- static const bool `value` = `sizeof(__test<T>(0)) == 1`

## Static Private Member Functions

- `template<typename U >`  
`static __one__test (_Wrap_type< typename U::iterator > *)`
- `template<typename U >`  
`static __two__test (...)`

## Additional Inherited Members

### 6.3.1 Detailed Description

```
template<typename T>
class Rcpp::traits::_has_iterator_helper< T >
```

Definition at line 40 of file `has_iterator.h`.

### 6.3.2 Member Function Documentation

#### 6.3.2.1 \_\_test() [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_has_iterator_helper< T >::__test (
    ... ) [static], [private]
```

#### 6.3.2.2 \_\_test() [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_has_iterator_helper< T >::__test (
    _Wrap_type< typename U::iterator > * ) [static], [private]
```

### 6.3.3 Member Data Documentation

### 6.3.3.1 value

```
template<typename T >
const bool Rcpp::traits::_has_iterator_helper< T >::value = sizeof(__test<T>(0)) == 1 [static]
```

Definition at line 50 of file `has_iterator.h`.

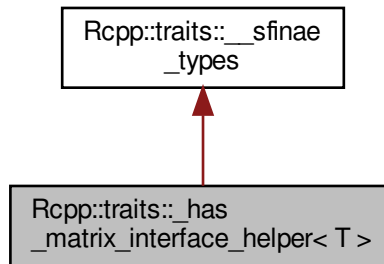
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/traits/has_iterator.h`

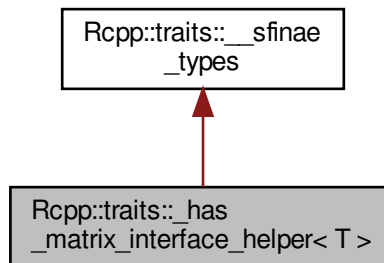
## 6.4 Rcpp::traits::\_has\_matrix\_interface\_helper< T > Class Template Reference

```
#include <matrix_interface.h>
```

Inheritance diagram for `Rcpp::traits::_has_matrix_interface_helper< T >`:



Collaboration diagram for `Rcpp::traits::_has_matrix_interface_helper< T >`:



## Classes

- struct [\\_Wrap\\_type](#)

## Static Public Attributes

- static const bool [value](#) = sizeof(\_\_test<T>(0)) == 1

## Static Private Member Functions

- `template<typename U >`  
`static \_\_one\_\_test (\_Wrap\_type< typename U::r_matrix_interface > *)`
- `template<typename U >`  
`static \_\_two\_\_test (...)`

## Additional Inherited Members

### 6.4.1 Detailed Description

```
template<typename T>
class Rcpp::traits::_has_matrix_interface_helper< T >
```

Definition at line 33 of file `matrix_interface.h`.

### 6.4.2 Member Function Documentation

#### 6.4.2.1 [\\_\\_test\(\)](#) [1/2]

```
template<typename T >
template<typename U >
static \_\_two Rcpp::traits::_has_matrix_interface_helper< T >::__test (
    ... ) [static], [private]
```

#### 6.4.2.2 [\\_\\_test\(\)](#) [2/2]

```
template<typename T >
template<typename U >
static \_\_one Rcpp::traits::_has_matrix_interface_helper< T >::__test (
    \_Wrap\_type< typename U::r_matrix_interface > * ) [static], [private]
```

### 6.4.3 Member Data Documentation

#### 6.4.3.1 value

```
template<typename T >
const bool Rcpp::traits::_has_matrix_interface_helper< T >::value = sizeof(__test<T>(0)) == 1
[static]
```

Definition at line 43 of file `matrix_interface.h`.

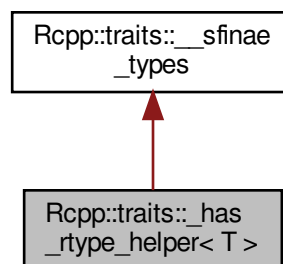
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/traits/matrix_interface.h`

## 6.5 Rcpp::traits::\_has\_rtype\_helper< T > Class Template Reference

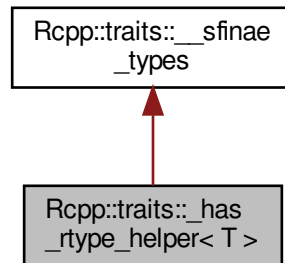
```
#include <expands_to_logical.h>
```

Inheritance diagram for `Rcpp::traits::_has_rtype_helper< T >`:





Collaboration diagram for Rcpp::traits::\_has\_rtype\_helper< T >:



## Classes

- struct [\\_Wrap\\_type](#)

## Static Public Attributes

- static const bool [value](#) = sizeof([\\_\\_test](#)<T>(0)) == 1

## Static Private Member Functions

- template<typename U >  
static [\\_\\_one \\_\\_test](#) ([\\_Wrap\\_type](#)< typename U::r\_expands\_to\_logical > \*)
- template<typename U >  
static [\\_\\_two \\_\\_test](#) (...)

## Additional Inherited Members

### 6.5.1 Detailed Description

```
template<typename T>  
class Rcpp::traits::_has_rtype_helper< T >
```

Definition at line 41 of file `expands_to_logical.h`.

### 6.5.2 Member Function Documentation

### 6.5.2.1 `__test()` [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_has_rtype_helper< T >::__test (
    ... ) [static], [private]
```

### 6.5.2.2 `__test()` [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_has_rtype_helper< T >::__test (
    _Wrap_type< typename U::r_expands_to_logical > * ) [static], [private]
```

## 6.5.3 Member Data Documentation

### 6.5.3.1 `value`

```
template<typename T >
const bool Rcpp::traits::_has_rtype_helper< T >::value = sizeof(__test<T>(0)) == 1 [static]
```

Definition at line 51 of file `expands_to_logical.h`.

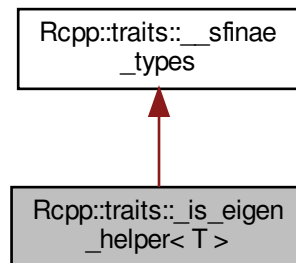
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/traits/expands_to_logical.h`

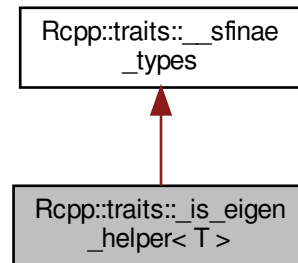
## 6.6 Rcpp::traits::\_is\_eigen\_helper< T > Class Template Reference

```
#include <is_eigen_base.h>
```

Inheritance diagram for Rcpp::traits::\_is\_eigen\_helper< T >:



Collaboration diagram for Rcpp::traits::\_is\_eigen\_helper< T >:



## Classes

- struct [\\_Wrap\\_type](#)

## Static Public Attributes

- static const bool [value](#) = sizeof([\\_\\_test](#)<T>(0)) == 1

## Static Private Member Functions

- template<typename U >  
static [\\_\\_one \\_\\_test](#) ([\\_Wrap\\_type](#)< typename U::StorageKind > \*)
- template<typename U >  
static [\\_\\_two \\_\\_test](#) (...)

## Additional Inherited Members

### 6.6.1 Detailed Description

```
template<typename T>  
class Rcpp::traits::_is_eigen_helper< T >
```

Definition at line 29 of file `is_eigen_base.h`.

### 6.6.2 Member Function Documentation

### 6.6.2.1 `__test()` [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_is_eigen_helper< T >::__test (
    ... ) [static], [private]
```

### 6.6.2.2 `__test()` [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_is_eigen_helper< T >::__test (
    _Wrap_type< typename U::StorageKind > * ) [static], [private]
```

## 6.6.3 Member Data Documentation

### 6.6.3.1 `value`

```
template<typename T >
const bool Rcpp::traits::_is_eigen_helper< T >::value = sizeof(__test<T>(0)) == 1 [static]
```

Definition at line 39 of file `is_eigen_base.h`.

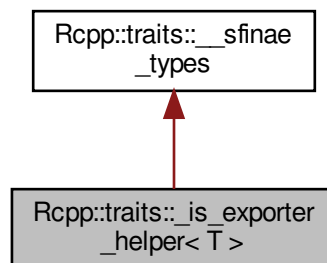
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/traits/is\\_eigen\\_base.h](#)

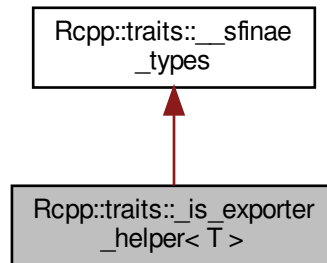
## 6.7 Rcpp::traits::\_is\_exporter\_helper< T > Class Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::\_is\_exporter\_helper< T >:



Collaboration diagram for Rcpp::traits::\_is\_exporter\_helper< T >:



## Classes

- struct [\\_Wrap\\_type](#)

## Static Public Attributes

- static const bool [value](#) = sizeof([\\_\\_test](#)<T>(0)) == 1

## Static Private Member Functions

- `template<typename U >`  
`static \_\_one \_\_test (\_Wrap\_type< typename U::r_export_type > *)`
- `template<typename U >`  
`static \_\_two \_\_test (...)`

## Additional Inherited Members

### 6.7.1 Detailed Description

```

template<typename T>
class Rcpp::traits::_is_exporter_helper< T >

```

Definition at line 83 of file `has_iterator.h`.

### 6.7.2 Member Function Documentation

### 6.7.2.1 `__test()` [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_is_exporter_helper< T >::__test (
    ... ) [static], [private]
```

### 6.7.2.2 `__test()` [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_is_exporter_helper< T >::__test (
    _Wrap_type< typename U::r_export_type > * ) [static], [private]
```

## 6.7.3 Member Data Documentation

### 6.7.3.1 `value`

```
template<typename T >
const bool Rcpp::traits::_is_exporter_helper< T >::value = sizeof(__test<T>(0)) == 1 [static]
```

Definition at line 93 of file `has_iterator.h`.

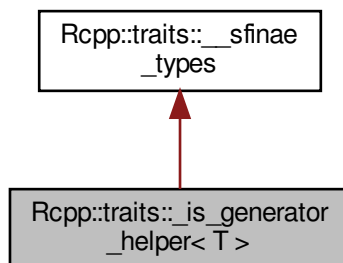
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/traits/has_iterator.h`

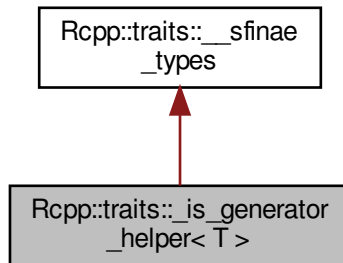
## 6.8 Rcpp::traits::\_is\_generator\_helper< T > Class Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::\_is\_generator\_helper< T >:



Collaboration diagram for Rcpp::traits::\_is\_generator\_helper< T >:



## Classes

- struct [\\_Wrap\\_type](#)

## Static Public Attributes

- static const bool [value](#) = sizeof([\\_\\_test](#)<T>(0)) == 1

## Static Private Member Functions

- template<typename U >  
static [\\_\\_one \\_\\_test](#) ([\\_Wrap\\_type](#)< typename U::r\_generator > \*)
- template<typename U >  
static [\\_\\_two \\_\\_test](#) (...)

## Additional Inherited Members

### 6.8.1 Detailed Description

```
template<typename T>  
class Rcpp::traits::_is_generator_helper< T >
```

Definition at line 68 of file `has_iterator.h`.

### 6.8.2 Member Function Documentation

**6.8.2.1** `__test()` [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_is_generator_helper< T >::__test (
    ... ) [static], [private]
```

**6.8.2.2** `__test()` [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_is_generator_helper< T >::__test (
    _Wrap_type< typename U::r_generator > * ) [static], [private]
```

**6.8.3 Member Data Documentation****6.8.3.1 value**

```
template<typename T >
const bool Rcpp::traits::_is_generator_helper< T >::value = sizeof(__test<T>(0)) == 1 [static]
```

Definition at line 78 of file `has_iterator.h`.

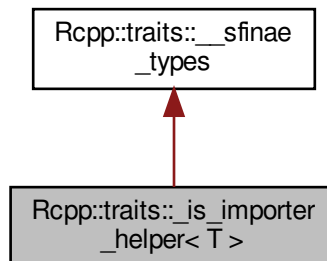
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/traits/has_iterator.h`

**6.9 Rcpp::traits::\_is\_importer\_helper< T > Class Template Reference**

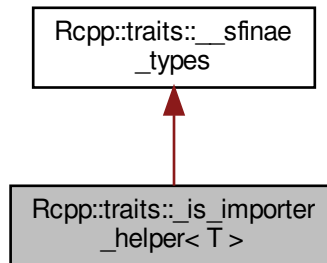
```
#include <has_iterator.h>
```

Inheritance diagram for `Rcpp::traits::_is_importer_helper< T >`:





Collaboration diagram for Rcpp::traits::\_is\_importer\_helper< T >:



## Classes

- struct [\\_Wrap\\_type](#)

## Static Public Attributes

- static const bool [value](#) = sizeof([\\_\\_test](#)<T>(0)) == 1

## Static Private Member Functions

- template<typename U >  
static [\\_\\_one \\_\\_test](#) ([\\_Wrap\\_type](#)< typename U::r\_import\_type > \*)
- template<typename U >  
static [\\_\\_two \\_\\_test](#) (...)

## Additional Inherited Members

### 6.9.1 Detailed Description

```
template<typename T>  
class Rcpp::traits::_is_importer_helper< T >
```

Definition at line 54 of file `has_iterator.h`.

### 6.9.2 Member Function Documentation

### 6.9.2.1 `__test()` [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_is_importer_helper< T >::__test (
    ... ) [static], [private]
```

### 6.9.2.2 `__test()` [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_is_importer_helper< T >::__test (
    _Wrap_type< typename U::r_import_type > * ) [static], [private]
```

## 6.9.3 Member Data Documentation

### 6.9.3.1 `value`

```
template<typename T >
const bool Rcpp::traits::_is_importer_helper< T >::value = sizeof(__test<T>(0)) == 1 [static]
```

Definition at line 64 of file `has_iterator.h`.

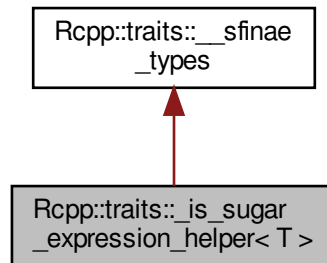
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/traits/has_iterator.h`

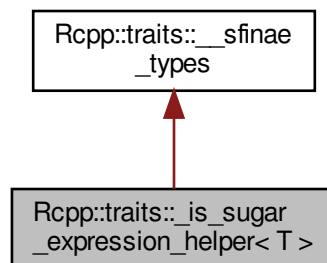
## 6.10 `Rcpp::traits::_is_sugar_expression_helper`< T > Class Template Reference

```
#include <is_sugar_expression.h>
```

Inheritance diagram for Rcpp::traits::\_is\_sugar\_expression\_helper< T >:



Collaboration diagram for Rcpp::traits::\_is\_sugar\_expression\_helper< T >:



## Classes

- struct [\\_Wrap\\_type](#)

## Static Public Attributes

- static const bool [value](#) = sizeof([\\_\\_test](#)<T>(0)) == 1

## Static Private Member Functions

- template<typename U >  
static [\\_\\_one \\_\\_test](#) ([\\_Wrap\\_type](#)< typename U::rcpp\_sugar\_expression > \*)
- template<typename U >  
static [\\_\\_two \\_\\_test](#) (...)

## Additional Inherited Members

### 6.10.1 Detailed Description

```
template<typename T>
class Rcpp::traits::_is_sugar_expression_helper< T >
```

Definition at line 29 of file `is_sugar_expression.h`.

### 6.10.2 Member Function Documentation

#### 6.10.2.1 `__test()` [1/2]

```
template<typename T >
template<typename U >
static __two Rcpp::traits::_is_sugar_expression_helper< T >::__test (
    ... ) [static], [private]
```

#### 6.10.2.2 `__test()` [2/2]

```
template<typename T >
template<typename U >
static __one Rcpp::traits::_is_sugar_expression_helper< T >::__test (
    _Wrap_type< typename U::rcpp_sugar_expression > * ) [static], [private]
```

### 6.10.3 Member Data Documentation

#### 6.10.3.1 `value`

```
template<typename T >
const bool Rcpp::traits::_is_sugar_expression_helper< T >::value = sizeof(__test<T>(0)) == 1
[static]
```

Definition at line 39 of file `is_sugar_expression.h`.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/traits/is\\_sugar\\_expression.h](#)

## 6.11 Rcpp::traits::\_has\_iterator\_helper< T >::\_Wrap\_type< U > Struct Template Reference

### 6.11.1 Detailed Description

```
template<typename T>
template<typename U>
struct Rcpp::traits::_has_iterator_helper< T >::_Wrap_type< U >
```

Definition at line 41 of file `has_iterator.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/has_iterator.h`

## 6.12 Rcpp::traits::\_has\_matrix\_interface\_helper< T >::\_Wrap\_type< U > Struct Template Reference

### 6.12.1 Detailed Description

```
template<typename T>
template<typename U>
struct Rcpp::traits::_has_matrix_interface_helper< T >::_Wrap_type< U >
```

Definition at line 34 of file `matrix_interface.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/matrix_interface.h`

## 6.13 Rcpp::traits::\_has\_rtype\_helper< T >::\_Wrap\_type< U > Struct Template Reference

### 6.13.1 Detailed Description

```
template<typename T>
template<typename U>
struct Rcpp::traits::_has_rtype_helper< T >::_Wrap_type< U >
```

Definition at line 42 of file `expands_to_logical.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/expands_to_logical.h`

## 6.14 Rcpp::traits::\_is\_eigen\_helper< T >::\_Wrap\_type< U > Struct Template Reference

### 6.14.1 Detailed Description

```
template<typename T>
template<typename U>
struct Rcpp::traits::_is_eigen_helper< T >::_Wrap_type< U >
```

Definition at line 30 of file `is_eigen_base.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_eigen\\_base.h](#)

## 6.15 Rcpp::traits::\_is\_exporter\_helper< T >::\_Wrap\_type< U > Struct Template Reference

### 6.15.1 Detailed Description

```
template<typename T>
template<typename U>
struct Rcpp::traits::_is_exporter_helper< T >::_Wrap_type< U >
```

Definition at line 84 of file `has_iterator.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/has\\_iterator.h](#)

## 6.16 Rcpp::traits::\_is\_generator\_helper< T >::\_Wrap\_type< U > Struct Template Reference

### 6.16.1 Detailed Description

```
template<typename T>
template<typename U>
struct Rcpp::traits::_is_generator_helper< T >::_Wrap_type< U >
```

Definition at line 69 of file `has_iterator.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/has\\_iterator.h](#)

## 6.17 Rcpp::traits::\_is\_importer\_helper< T >::\_Wrap\_type< U > Struct Template Reference

### 6.17.1 Detailed Description

```
template<typename T>
template<typename U>
struct Rcpp::traits::_is_importer_helper< T >::_Wrap_type< U >
```

Definition at line 55 of file `has_iterator.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/has\\_iterator.h](#)

## 6.18 Rcpp::traits::\_is\_sugar\_expression\_helper< T >::\_Wrap\_type< U > Struct Template Reference

### 6.18.1 Detailed Description

```
template<typename T>
template<typename U>
struct Rcpp::traits::_is_sugar_expression_helper< T >::_Wrap_type< U >
```

Definition at line 30 of file `is_sugar_expression.h`.

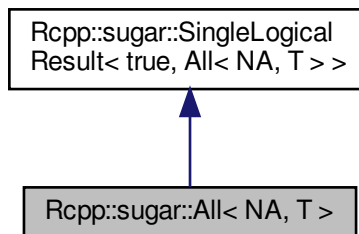
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_sugar\\_expression.h](#)

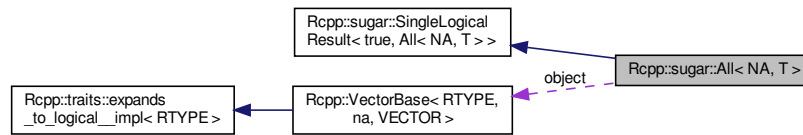
## 6.19 Rcpp::sugar::All< NA, T > Class Template Reference

```
#include <all.h>
```

Inheritance diagram for `Rcpp::sugar::All< NA, T >`:



Collaboration diagram for `Rcpp::sugar::All< NA, T >`:



## Public Types

- typedef `Rcpp::VectorBase< LGLSXP, NA, T >` `VEC_TYPE`
- typedef `SingleLogicalResult< true, All< NA, T > >` `PARENT`

## Public Member Functions

- `All` (const `VEC_TYPE` &t)
- void `apply` ()

## Private Attributes

- const `VEC_TYPE` & `object`

## Additional Inherited Members

### 6.19.1 Detailed Description

```

template<bool NA, typename T>
class Rcpp::sugar::All< NA, T >

```

Definition at line 29 of file `all.h`.

### 6.19.2 Member Typedef Documentation

#### 6.19.2.1 PARENT

```

template<bool NA, typename T >
typedef SingleLogicalResult< true, All<NA,T> > Rcpp::sugar::All< NA, T >::PARENT

```

Definition at line 32 of file `all.h`.



### 6.19.2.2 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<LGLSXP,NA,T> Rcpp::sugar::All< NA, T >::VEC_TYPE
```

Definition at line 31 of file all.h.

## 6.19.3 Constructor & Destructor Documentation

### 6.19.3.1 All()

```
template<bool NA, typename T >
Rcpp::sugar::All< NA, T >::All (
    const VEC_TYPE & t ) [inline]
```

Definition at line 33 of file all.h.

## 6.19.4 Member Function Documentation

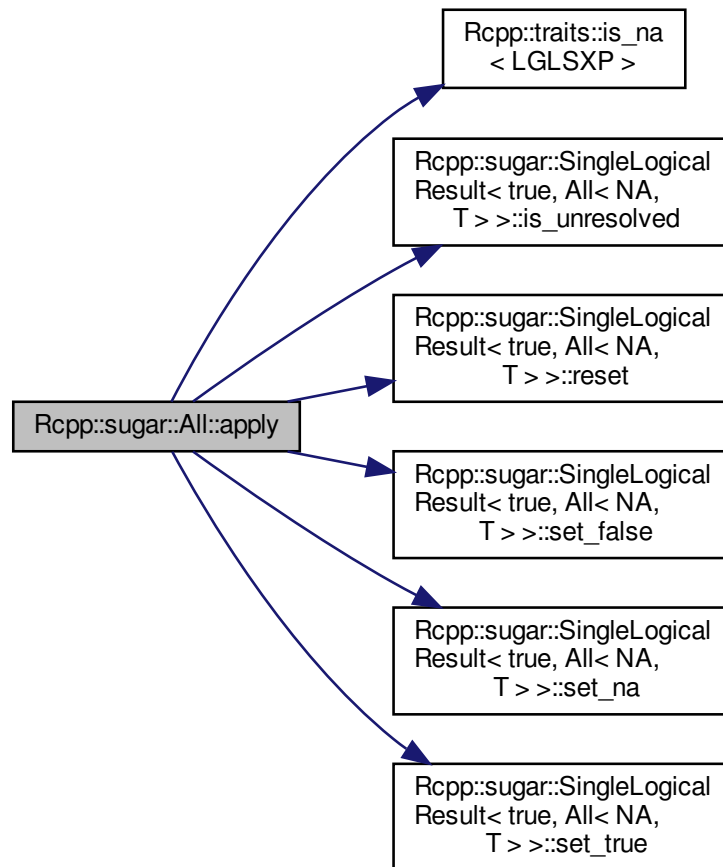
### 6.19.4.1 apply()

```
template<bool NA, typename T >
void Rcpp::sugar::All< NA, T >::apply ( ) [inline]
```

Definition at line 35 of file all.h.

References `Rcpp::traits::is_na< LGLSXP >()`, `Rcpp::sugar::SingleLogicalResult< true, All< NA, T > >::is_unresolved()`, `Rcpp::sugar::SingleLogicalResult< true, All< NA, T > >::reset()`, `Rcpp::sugar::SingleLogicalResult< true, All< NA, T > >::set_false()`, `Rcpp::sugar::SingleLogicalResult< true, All< NA, T > >::set_na()`, and `Rcpp::sugar::SingleLogicalResult< true, All< NA, T > >::set_true()`.

Here is the call graph for this function:



## 6.19.5 Member Data Documentation

### 6.19.5.1 object

```

template<bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::All< NA, T >::object [private]
  
```

Definition at line 54 of file all.h.

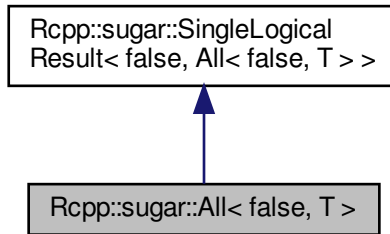
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/all.h`

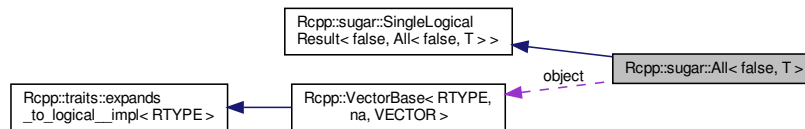
## 6.20 Rcpp::sugar::All< false, T > Class Template Reference

```
#include <all.h>
```

Inheritance diagram for Rcpp::sugar::All< false, T >:



Collaboration diagram for Rcpp::sugar::All< false, T >:



### Public Types

- typedef `Rcpp::VectorBase< LGLSXP, false, T >` `VEC_TYPE`
- typedef `SingleLogicalResult< false, All< false, T > >` `PARENT`

### Public Member Functions

- `All` (const `VEC_TYPE` &t)
- void `apply` ()

### Private Attributes

- const `VEC_TYPE` & `object`

## Additional Inherited Members

### 6.20.1 Detailed Description

```
template<typename T>  
class Rcpp::sugar::All< false, T >
```

Definition at line 60 of file all.h.

### 6.20.2 Member Typedef Documentation

#### 6.20.2.1 PARENT

```
template<typename T >  
typedef SingleLogicalResult< false, All<false,T> > Rcpp::sugar::All< false, T >::PARENT
```

Definition at line 63 of file all.h.

#### 6.20.2.2 VEC\_TYPE

```
template<typename T >  
typedef Rcpp::VectorBase<LGLSXP,false,T> Rcpp::sugar::All< false, T >::VEC_TYPE
```

Definition at line 62 of file all.h.

### 6.20.3 Constructor & Destructor Documentation

#### 6.20.3.1 All()

```
template<typename T >  
Rcpp::sugar::All< false, T >::All (  
    const VEC\_TYPE & t ) [inline]
```

Definition at line 64 of file all.h.

## 6.20.4 Member Function Documentation

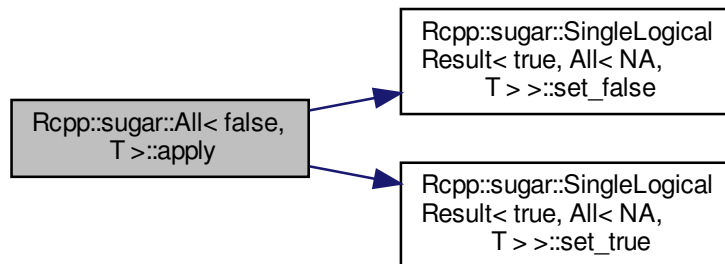
### 6.20.4.1 apply()

```
template<typename T >
void Rcpp::sugar::All< false, T >::apply ( ) [inline]
```

Definition at line 66 of file all.h.

References `Rcpp::sugar::SingleLogicalResult< true, All< NA, T > >::set_false()`, and `Rcpp::sugar::SingleLogicalResult< true, All< NA, T > >::set_true()`.

Here is the call graph for this function:



## 6.20.5 Member Data Documentation

### 6.20.5.1 object

```
template<typename T >
const VEC_TYPE& Rcpp::sugar::All< false, T >::object [private]
```

Definition at line 77 of file all.h.

The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/all.h`

## 6.21 Rcpp::traits::allowed\_matrix\_type< bool > Struct Template Reference

### 6.21.1 Detailed Description

```
template<bool>
struct Rcpp::traits::allowed_matrix_type< bool >
```

Definition at line 29 of file one\_type.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/one\\_type.h](#)

## 6.22 Rcpp::traits::allowed\_matrix\_type< true > Struct Reference

```
#include <one_type.h>
```

### 6.22.1 Detailed Description

Definition at line 32 of file one\_type.h.

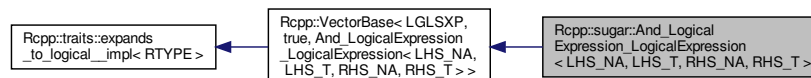
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/one\\_type.h](#)

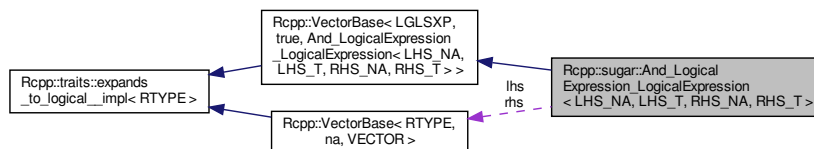
## 6.23 Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< LGLSXP, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)

## Public Member Functions

- [And\\_LogicalExpression\\_LogicalExpression](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- int [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_TYPE](#) & lhs
- const [RHS\\_TYPE](#) & rhs

### 6.23.1 Detailed Description

```

template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >
    
```

Definition at line 205 of file and.h.

### 6.23.2 Member Typedef Documentation

### 6.23.2.1 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,LHS_NA,LHS_T> Rcpp::sugar::And_LogicalExpression_LogicalExpression<
LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 207 of file and.h.

### 6.23.2.2 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,RHS_NA,RHS_T> Rcpp::sugar::And_LogicalExpression_LogicalExpression<
LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 208 of file and.h.

## 6.23.3 Constructor & Destructor Documentation

### 6.23.3.1 And\_LogicalExpression\_LogicalExpression()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::And_LogicalExpression_LogicalExpression(
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 210 of file and.h.

## 6.23.4 Member Function Documentation

### 6.23.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
int Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
(
    R_xlen_t i ) const [inline]
```

Definition at line 212 of file and.h.

References [Rcpp::sugar::And\\_LogicalExpression\\_LogicalExpression< LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#), and [Rcpp::sugar::And\\_LogicalExpression\\_LogicalExpression< LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::rhs](#).



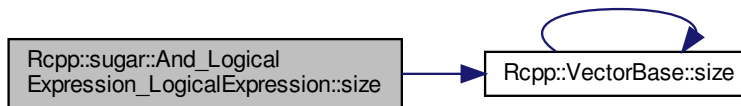
### 6.23.4.2 size()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >↵
::size ( ) const [inline]
```

Definition at line 217 of file and.h.

References Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



## 6.23.5 Member Data Documentation

### 6.23.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS↵
_T >::lhs [private]
```

Definition at line 220 of file and.h.

Referenced by Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_↵  
T >::operator[](), Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< false, LHS\_T, RHS\_NA, RHS\_↵  
T >::operator[](), Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, false, RHS\_T >↵  
::operator[](), Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< false, LHS\_T, false, RHS\_T >↵  
::operator[](), Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >↵  
::size(), Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< false, LHS\_T, RHS\_NA, RHS\_T >↵  
::size(), Rcpp::sugar::And\_↵  
\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, false, RHS\_T >↵  
::size(), and Rcpp::sugar::And\_Logical\_↵  
Expression\_LogicalExpression< false, LHS\_T, false, RHS\_T >↵  
::size().

### 6.23.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs [private]
```

Definition at line 221 of file and.h.

Referenced by `Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[]()`, `Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::operator[]()`, `Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::operator[]()`, and `Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::operator[]()`.

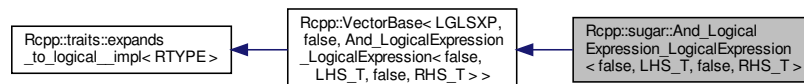
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/logical/and.h](#)

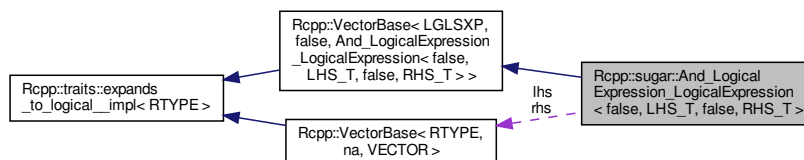
## 6.24 Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< false, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for `Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >`:



Collaboration diagram for `Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >`:



## Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< LGLSXP, false, RHS\_T > [RHS\\_TYPE](#)

## Public Member Functions

- [And\\_LogicalExpression\\_LogicalExpression](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- int [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_TYPE](#) & lhs
- const [RHS\\_TYPE](#) & rhs

### 6.24.1 Detailed Description

```
template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >
```

Definition at line 264 of file and.h.

### 6.24.2 Member Typedef Documentation

#### 6.24.2.1 LHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP, false, LHS_T> Rcpp::sugar::And\_LogicalExpression\_LogicalExpression<
false, LHS_T, false, RHS_T >::LHS\_TYPE
```

Definition at line 267 of file and.h.

#### 6.24.2.2 RHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP, false, RHS_T> Rcpp::sugar::And\_LogicalExpression\_LogicalExpression<
false, LHS_T, false, RHS_T >::RHS\_TYPE
```

Definition at line 268 of file and.h.

## 6.24.3 Constructor & Destructor Documentation

### 6.24.3.1 And\_LogicalExpression\_LogicalExpression()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::And_LogicalExpression_LogicalExpression(
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 270 of file and.h.

## 6.24.4 Member Function Documentation

### 6.24.4.1 operator[]()

```
template<typename LHS_T , typename RHS_T >
int Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::operator[]
(
    R_xlen_t i ) const [inline]
```

Definition at line 272 of file and.h.

References `Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

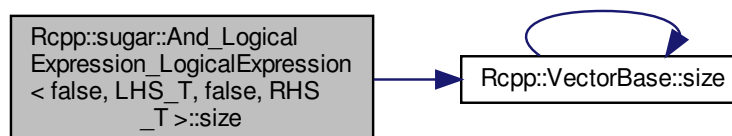
### 6.24.4.2 size()

```
template<typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::size
( ) const [inline]
```

Definition at line 276 of file and.h.

References `Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`.

Here is the call graph for this function:



### 6.24.5 Member Data Documentation

#### 6.24.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T
>::lhs [private]
```

Definition at line 279 of file and.h.

#### 6.24.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T
>::rhs [private]
```

Definition at line 280 of file and.h.

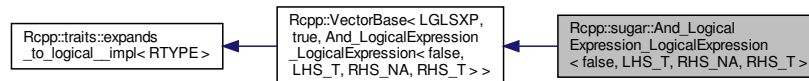
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

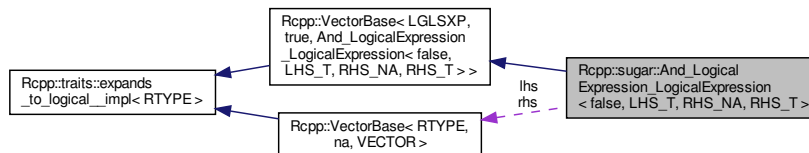
## 6.25 Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< false, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< false, LHS\_T, RHS\_NA, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< LGLSXP, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)

## Public Member Functions

- [And\\_LogicalExpression\\_LogicalExpression](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- int [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_TYPE](#) & lhs
- const [RHS\\_TYPE](#) & rhs

### 6.25.1 Detailed Description

```
template<typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >
```

Definition at line 224 of file and.h.

### 6.25.2 Member Typedef Documentation

#### 6.25.2.1 LHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP, false, LHS_T> Rcpp::sugar::And\_LogicalExpression\_LogicalExpression<
false, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 227 of file and.h.

#### 6.25.2.2 RHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP, RHS_NA, RHS_T> Rcpp::sugar::And\_LogicalExpression\_LogicalExpression<
false, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 228 of file and.h.

## 6.25.3 Constructor & Destructor Documentation

### 6.25.3.1 And\_LogicalExpression\_LogicalExpression()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::And_LogicalExpression_LogicalExpression  
(  
    const LHS_TYPE & lhs_  
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 230 of file and.h.

## 6.25.4 Member Function Documentation

### 6.25.4.1 operator[]()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
int Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::operator[]  
(  
    R_xlen_t i ) const [inline]
```

Definition at line 232 of file and.h.

References Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

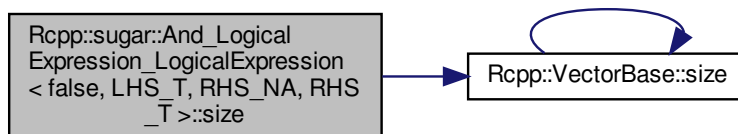
### 6.25.4.2 size()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
R_xlen_t Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::size ( ) const [inline]
```

Definition at line 237 of file and.h.

References Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



## 6.25.5 Member Data Documentation

### 6.25.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T
>::lhs [private]
```

Definition at line 240 of file and.h.

### 6.25.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T
>::rhs [private]
```

Definition at line 241 of file and.h.

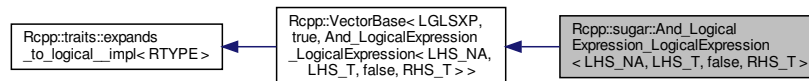
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/logical/and.h](#)

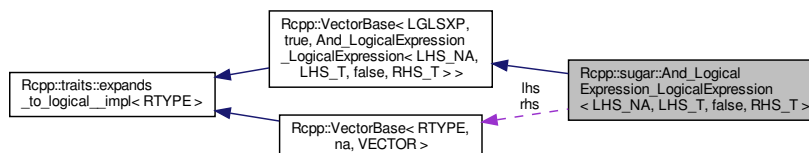
## 6.26 Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, false, RHS\_T >:





## Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< LGLSXP, false, RHS\_T > [RHS\\_TYPE](#)

## Public Member Functions

- [And\\_LogicalExpression\\_LogicalExpression](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- int [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_TYPE](#) &lhs
- const [RHS\\_TYPE](#) &rhs

### 6.26.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, typename RHS_T>  
class Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >
```

Definition at line 244 of file and.h.

### 6.26.2 Member Typedef Documentation

#### 6.26.2.1 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef Rcpp::VectorBase<LGLSXP, LHS_NA, LHS_T> Rcpp::sugar::And\_LogicalExpression\_LogicalExpression<  
LHS_NA, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 247 of file and.h.

#### 6.26.2.2 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef Rcpp::VectorBase<LGLSXP, false, RHS_T> Rcpp::sugar::And\_LogicalExpression\_LogicalExpression<  
LHS_NA, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 248 of file and.h.

## 6.26.3 Constructor & Destructor Documentation

### 6.26.3.1 And\_LogicalExpression\_LogicalExpression()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::And_LogicalExpression_LogicalExpression(
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 250 of file and.h.

## 6.26.4 Member Function Documentation

### 6.26.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
int Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::operator[]
(
    R_xlen_t i ) const [inline]
```

Definition at line 252 of file and.h.

References `Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

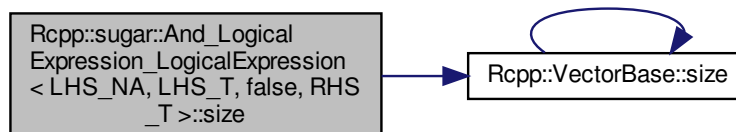
### 6.26.4.2 size()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::size
( ) const [inline]
```

Definition at line 257 of file and.h.

References `Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`.

Here is the call graph for this function:



## 6.26.5 Member Data Documentation

### 6.26.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T
>::lhs [private]
```

Definition at line 260 of file and.h.

### 6.26.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T
>::rhs [private]
```

Definition at line 261 of file and.h.

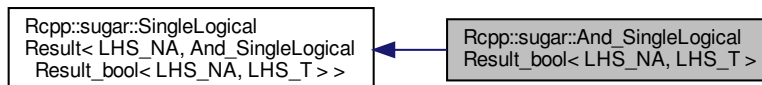
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/[and.h](#)

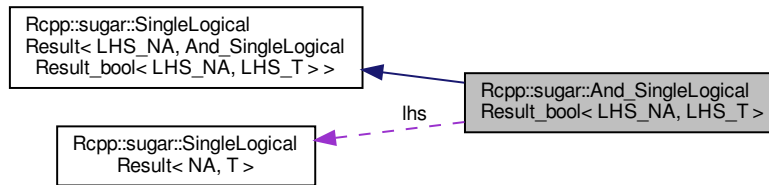
## 6.27 Rcpp::sugar::And\_SingleLogicalResult\_bool< LHS\_NA, LHS\_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And\_SingleLogicalResult\_bool< LHS\_NA, LHS\_T >:



Collaboration diagram for `Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >`:



## Public Types

- typedef `SingleLogicalResult< LHS_NA, LHS_T >` `LHS_TYPE`
- typedef `SingleLogicalResult< LHS_NA, And_SingleLogicalResult_bool< LHS_NA, LHS_T > >` `BASE`

## Public Member Functions

- `And_SingleLogicalResult_bool` (const `LHS_TYPE` &lhs\_, bool rhs\_)
- void `apply` ()

## Private Attributes

- const `LHS_TYPE` &lhs
- bool rhs

## Additional Inherited Members

### 6.27.1 Detailed Description

```

template<bool LHS_NA, typename LHS_T>
class Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >
  
```

Definition at line 171 of file and.h.

### 6.27.2 Member Typedef Documentation

### 6.27.2.1 BASE

```
template<bool LHS_NA, typename LHS_T >
typedef SingleLogicalResult< LHS_NA , And_SingleLogicalResult_bool<LHS_NA,LHS_T> > Rcpp::sugar::And_SingleLogicalResult_bool<
LHS_NA, LHS_T >::BASE
```

Definition at line 182 of file and.h.

### 6.27.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T >
typedef SingleLogicalResult<LHS_NA,LHS_T> Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA,
LHS_T >::LHS_TYPE
```

Definition at line 178 of file and.h.

## 6.27.3 Constructor & Destructor Documentation

### 6.27.3.1 And\_SingleLogicalResult\_bool()

```
template<bool LHS_NA, typename LHS_T >
Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::And_SingleLogicalResult_bool (
    const LHS_TYPE & lhs_,
    bool rhs_ ) [inline]
```

Definition at line 184 of file and.h.

### 6.27.4 Member Function Documentation

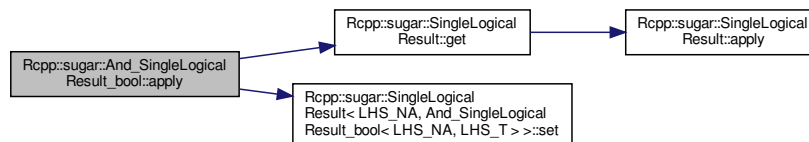
### 6.27.4.1 apply()

```
template<bool LHS_NA, typename LHS_T >
void Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply ( ) [inline]
```

Definition at line 187 of file and.h.

References `Rcpp::sugar::SingleLogicalResult< NA, T >::get()`, `Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::lhs`, `Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::rhs`, and `Rcpp::sugar::SingleLogicalResult< LHS_NA, And_SingleLogicalResult_bool< LHS_NA, LHS_T > >::set()`.

Here is the call graph for this function:



## 6.27.5 Member Data Documentation

### 6.27.5.1 lhs

```
template<bool LHS_NA, typename LHS_T >
const LHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::lhs [private]
```

Definition at line 196 of file and.h.

Referenced by `Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply()`.

### 6.27.5.2 rhs

```
template<bool LHS_NA, typename LHS_T >
bool Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::rhs [private]
```

Definition at line 197 of file and.h.

Referenced by `Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply()`.

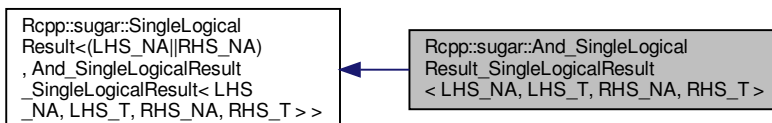
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/logical/and.h`

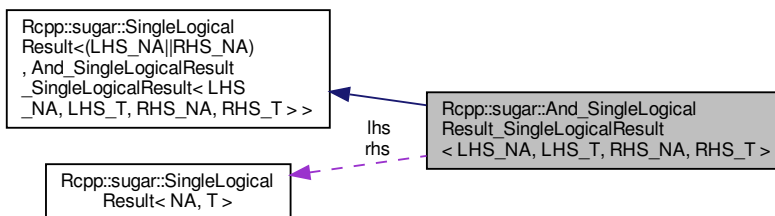
## 6.28 Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [SingleLogicalResult< LHS\\_NA, LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [SingleLogicalResult< RHS\\_NA, RHS\\_T >](#) [RHS\\_TYPE](#)
- typedef [SingleLogicalResult<\(LHS\\_NA||RHS\\_NA\), And\\_SingleLogicalResult\\_SingleLogicalResult< LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#) [BASE](#)

### Public Member Functions

- [And\\_SingleLogicalResult\\_SingleLogicalResult](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- void [apply](#) ()

## Private Attributes

- const [LHS\\_TYPE](#) & lhs
- const [RHS\\_TYPE](#) & rhs

## Additional Inherited Members

### 6.28.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 29 of file and.h.

### 6.28.2 Member Typedef Documentation

#### 6.28.2.1 BASE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult< (LHS_NA || RHS_NA) , And\_SingleLogicalResult\_SingleLogicalResult<LHS↔
_NA, LHS_T, RHS_NA, RHS_T> > Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS_NA, LHS↔
_T, RHS_NA, RHS_T >::BASE
```

Definition at line 41 of file and.h.

#### 6.28.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<LHS_NA, LHS_T> Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult<
LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS\_TYPE
```

Definition at line 36 of file and.h.

#### 6.28.2.3 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<RHS_NA, RHS_T> Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult<
LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS\_TYPE
```

Definition at line 37 of file and.h.



### 6.28.3 Constructor & Destructor Documentation

#### 6.28.3.1 And\_SingleLogicalResult\_SingleLogicalResult()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::And_SingleLogicalResult
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 43 of file and.h.

### 6.28.4 Member Function Documentation

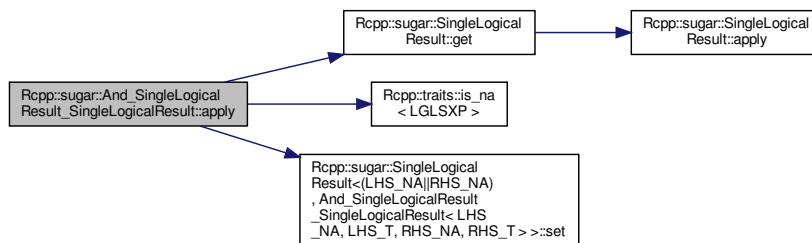
#### 6.28.4.1 apply()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
void Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::
::apply ( ) [inline]
```

Definition at line 46 of file and.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::traits::is\_na< LGLSXP >(), Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs, and Rcpp::sugar::SingleLogicalResult<(LHS\_NA||RHS\_NA), And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::set().

Here is the call graph for this function:



## 6.28.5 Member Data Documentation

### 6.28.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA,
RHS_T >::lhs [private]
```

Definition at line 58 of file and.h.

Referenced by `Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::apply()`, `Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::apply()`, `Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::apply()`, and `Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply()`.

### 6.28.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA,
RHS_T >::rhs [private]
```

Definition at line 59 of file and.h.

Referenced by `Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::apply()`, `Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::apply()`, `Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::apply()`, and `Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply()`.

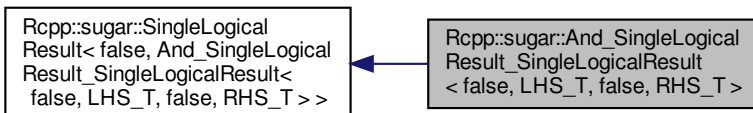
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/logical/and.h`

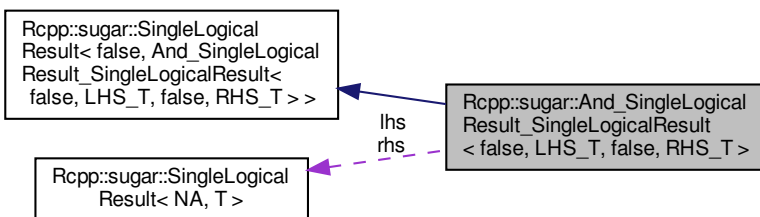
## 6.29 Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [SingleLogicalResult< false, LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [SingleLogicalResult< false, RHS\\_T >](#) [RHS\\_TYPE](#)
- typedef [SingleLogicalResult< false, And\\_SingleLogicalResult\\_SingleLogicalResult< false, LHS\\_T, false, RHS\\_T > >](#) [BASE](#)

### Public Member Functions

- [And\\_SingleLogicalResult\\_SingleLogicalResult](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- void [apply](#) ()

## Private Attributes

- const [LHS\\_TYPE](#) & lhs
- const [RHS\\_TYPE](#) & rhs

## Additional Inherited Members

### 6.29.1 Detailed Description

```
template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >
```

Definition at line 136 of file and.h.

### 6.29.2 Member Typedef Documentation

#### 6.29.2.1 BASE

```
template<typename LHS_T , typename RHS_T >
typedef SingleLogicalResult< false, And\_SingleLogicalResult\_SingleLogicalResult<false,LHS_T,
false,RHS_T> > Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< false, LHS_T, false,
RHS_T >::BASE
```

Definition at line 148 of file and.h.

#### 6.29.2.2 LHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef SingleLogicalResult<false,LHS_T> Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult<
false, LHS_T, false, RHS_T >::LHS\_TYPE
```

Definition at line 143 of file and.h.

#### 6.29.2.3 RHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef SingleLogicalResult<false,RHS_T> Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult<
false, LHS_T, false, RHS_T >::RHS\_TYPE
```

Definition at line 144 of file and.h.

## 6.29.3 Constructor & Destructor Documentation

### 6.29.3.1 And\_SingleLogicalResult\_SingleLogicalResult()

```
template<typename LHS_T , typename RHS_T >  
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::And_SingleLogicalResult_  
(  
    const LHS_TYPE & lhs_,  
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 150 of file and.h.

## 6.29.4 Member Function Documentation

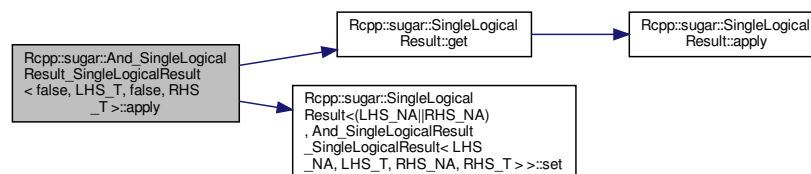
### 6.29.4.1 apply()

```
template<typename LHS_T , typename RHS_T >  
void Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >↔  
::apply ( ) [inline]
```

Definition at line 153 of file and.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs, and Rcpp::sugar::SingleLogicalResult<(LHS\_NA||RHS\_NA), And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > >::set().

Here is the call graph for this function:



## 6.29.5 Member Data Documentation

### 6.29.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false,
RHS_T >::lhs [private]
```

Definition at line 163 of file and.h.

### 6.29.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false,
RHS_T >::rhs [private]
```

Definition at line 164 of file and.h.

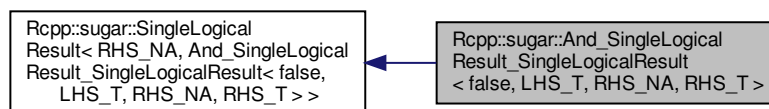
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/logical/and.h](#)

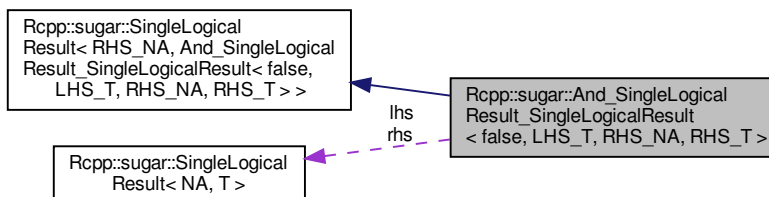
## 6.30 Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, RHS\_NA, RHS\_T >:



## Public Types

- typedef [SingleLogicalResult](#)< false, LHS\_T > [LHS\\_TYPE](#)
- typedef [SingleLogicalResult](#)< RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [SingleLogicalResult](#)< RHS\_NA, [And\\_SingleLogicalResult\\_SingleLogicalResult](#)< false, LHS\_T, RHS\_↵  
NA, RHS\_T > > [BASE](#)

## Public Member Functions

- [And\\_SingleLogicalResult\\_SingleLogicalResult](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- void [apply](#) ()

## Private Attributes

- const [LHS\\_TYPE](#) &lhs
- const [RHS\\_TYPE](#) &rhs

## Additional Inherited Members

### 6.30.1 Detailed Description

```
template<typename LHS_T, bool RHS_NA, typename RHS_T>  
class Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >
```

Definition at line 101 of file and.h.

### 6.30.2 Member Typedef Documentation

#### 6.30.2.1 BASE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef SingleLogicalResult< RHS_NA, And\_SingleLogicalResult\_SingleLogicalResult<false, LHS_↵  
T, RHS_NA, RHS_T> > Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< false, LHS_T, RHS_NA,  
RHS_T >::BASE
```

Definition at line 113 of file and.h.

### 6.30.2.2 LHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<false,LHS_T> Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 108 of file and.h.

### 6.30.2.3 RHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<RHS_NA,RHS_T> Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 109 of file and.h.

## 6.30.3 Constructor & Destructor Documentation

### 6.30.3.1 And\_SingleLogicalResult\_SingleLogicalResult()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::And_SingleLogicalResult_
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 115 of file and.h.

## 6.30.4 Member Function Documentation



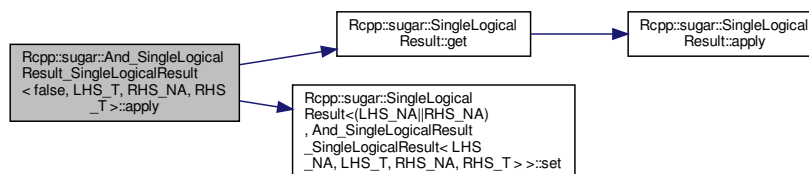
### 6.30.4.1 apply()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
void Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::
::apply ( ) [inline]
```

Definition at line 118 of file and.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs, and Rcpp::sugar::SingleLogicalResult<(LHS\_NA||RHS\_NA), And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > >::set().

Here is the call graph for this function:



## 6.30.5 Member Data Documentation

### 6.30.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA,
RHS_T >::lhs [private]
```

Definition at line 129 of file and.h.

### 6.30.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA,
RHS_T >::rhs [private]
```

Definition at line 130 of file and.h.

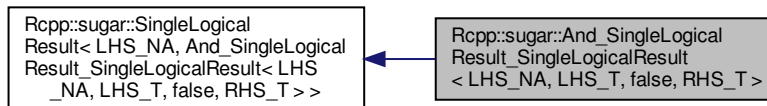
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/and.h

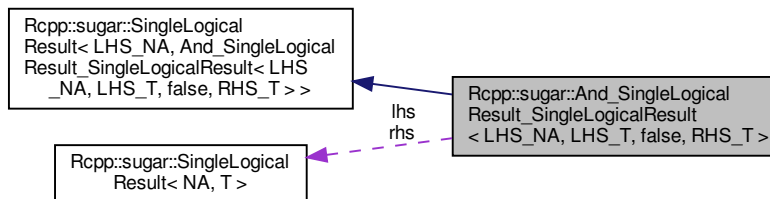
## 6.31 Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <and.h>
```

Inheritance diagram for Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [SingleLogicalResult< LHS\\_NA, LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [SingleLogicalResult< false, RHS\\_T >](#) [RHS\\_TYPE](#)
- typedef [SingleLogicalResult< LHS\\_NA, And\\_SingleLogicalResult\\_SingleLogicalResult< LHS\\_NA, LHS\\_T, false, RHS\\_T > >](#) [BASE](#)

### Public Member Functions

- [And\\_SingleLogicalResult\\_SingleLogicalResult](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- void [apply](#) ()

### Private Attributes

- const [LHS\\_TYPE](#) & [lhs](#)
- const [RHS\\_TYPE](#) & [rhs](#)

## Additional Inherited Members

### 6.31.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, typename RHS_T>  
class Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >
```

Definition at line 65 of file and.h.

### 6.31.2 Member Typedef Documentation

#### 6.31.2.1 BASE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef SingleLogicalResult< LHS_NA, And_SingleLogicalResult_SingleLogicalResult<LHS_NA,LHS_T,↵  
T,false,RHS_T> > Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false,  
RHS_T >::BASE
```

Definition at line 77 of file and.h.

#### 6.31.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef SingleLogicalResult<LHS_NA,LHS_T> Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<  
LHS_NA, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 72 of file and.h.

#### 6.31.2.3 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef SingleLogicalResult<false,RHS_T> Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<  
LHS_NA, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 73 of file and.h.

### 6.31.3 Constructor & Destructor Documentation

#### 6.31.3.1 And\_SingleLogicalResult\_SingleLogicalResult()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::And_SingleLogicalResult_
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 79 of file and.h.

### 6.31.4 Member Function Documentation

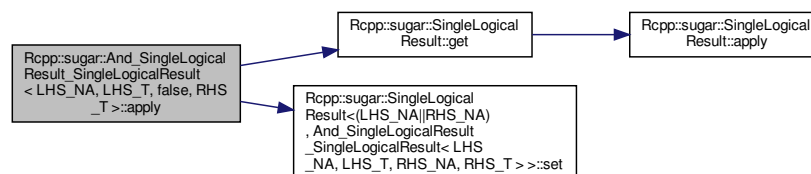
#### 6.31.4.1 apply()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
void Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >←
::apply ( ) [inline]
```

Definition at line 82 of file and.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs, and Rcpp::sugar::SingleLogicalResult<(LHS\_NA||RHS\_NA), And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > >::set().

Here is the call graph for this function:



### 6.31.5 Member Data Documentation

## 6.31.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false,
RHS_T >::lhs [private]
```

Definition at line 93 of file and.h.

## 6.31.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false,
RHS_T >::rhs [private]
```

Definition at line 94 of file and.h.

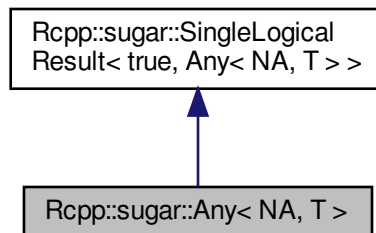
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/[and.h](#)

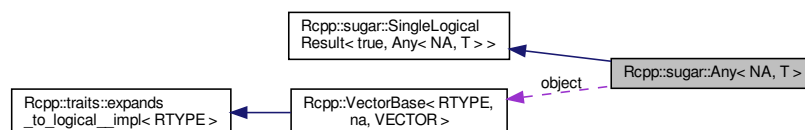
## 6.32 Rcpp::sugar::Any&lt; NA, T &gt; Class Template Reference

```
#include <any.h>
```

Inheritance diagram for Rcpp::sugar::Any< NA, T >:



Collaboration diagram for Rcpp::sugar::Any< NA, T >:



## Public Types

- typedef `Rcpp::VectorBase< LGLSXP, NA, T >` `VEC_TYPE`
- typedef `SingleLogicalResult< true, Any< NA, T > >` `PARENT`

## Public Member Functions

- `Any` (const `VEC_TYPE` &t)
- void `apply` ()

## Private Attributes

- const `VEC_TYPE` & `object`

## Additional Inherited Members

### 6.32.1 Detailed Description

```
template<bool NA, typename T>
class Rcpp::sugar::Any< NA, T >
```

Definition at line 29 of file any.h.

### 6.32.2 Member Typedef Documentation

#### 6.32.2.1 PARENT

```
template<bool NA, typename T >
typedef SingleLogicalResult< true , Any<NA,T> > Rcpp::sugar::Any< NA, T >::PARENT
```

Definition at line 32 of file any.h.

#### 6.32.2.2 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<LGLSXP,NA,T> Rcpp::sugar::Any< NA, T >::VEC_TYPE
```

Definition at line 31 of file any.h.

## 6.32.3 Constructor & Destructor Documentation

### 6.32.3.1 Any()

```
template<bool NA, typename T >  
Rcpp::sugar::Any< NA, T >::Any (  
    const VEC_TYPE & t ) [inline]
```

Definition at line 33 of file any.h.

## 6.32.4 Member Function Documentation

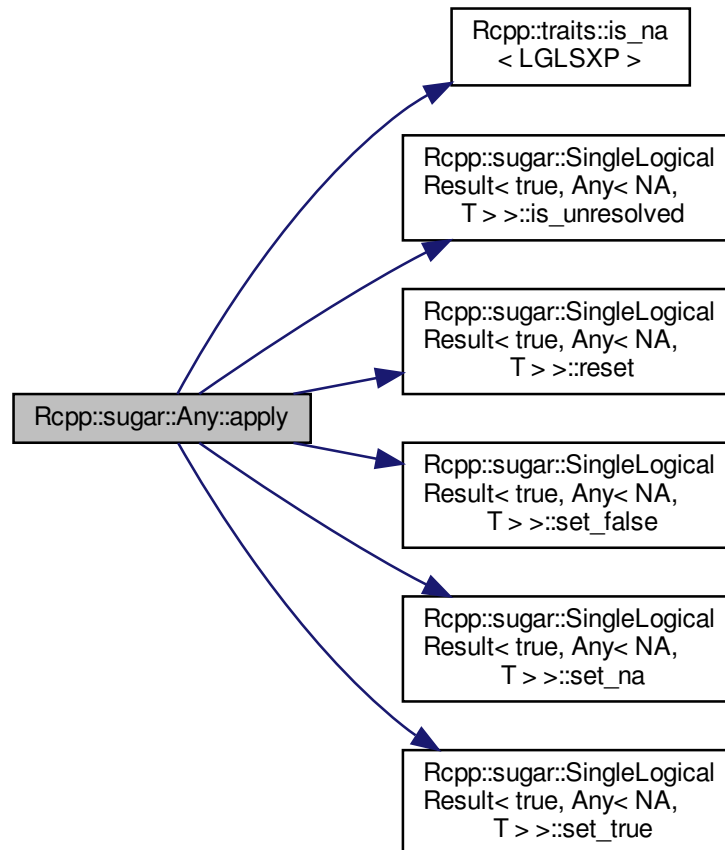
### 6.32.4.1 apply()

```
template<bool NA, typename T >  
void Rcpp::sugar::Any< NA, T >::apply ( ) [inline]
```

Definition at line 35 of file any.h.

References `Rcpp::traits::is_na< LGLSXP >()`, `Rcpp::sugar::SingleLogicalResult< true, Any< NA, T > >::is_↔ unresolved()`, `Rcpp::sugar::SingleLogicalResult< true, Any< NA, T > >::reset()`, `Rcpp::sugar::SingleLogicalResult< true, Any< NA, T > >::set_false()`, `Rcpp::sugar::SingleLogicalResult< true, Any< NA, T > >::set_na()`, and `Rcpp↔::sugar::SingleLogicalResult< true, Any< NA, T > >::set_true()`.

Here is the call graph for this function:



## 6.32.5 Member Data Documentation

### 6.32.5.1 object

```
template<bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Any< NA, T >::object [private]
```

Definition at line 54 of file `any.h`.

The documentation for this class was generated from the following file:

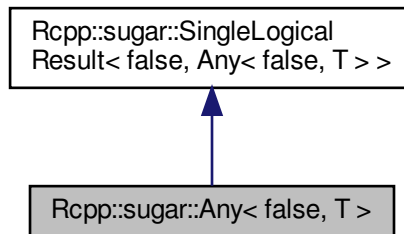
- `inst/include/Rcpp/sugar/functions/any.h`



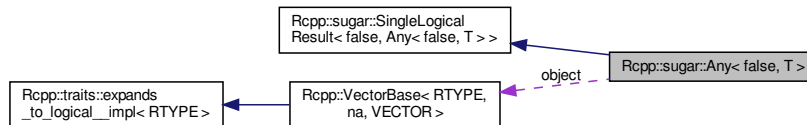
## 6.33 Rcpp::sugar::Any< false, T > Class Template Reference

```
#include <any.h>
```

Inheritance diagram for Rcpp::sugar::Any< false, T >:



Collaboration diagram for Rcpp::sugar::Any< false, T >:



### Public Types

- typedef `Rcpp::VectorBase< LGLSXP, false, T >` `VEC_TYPE`
- typedef `SingleLogicalResult< false, Any< false, T > >` `PARENT`

### Public Member Functions

- `Any` (const `VEC_TYPE` &t)
- void `apply` ()

### Private Attributes

- const `VEC_TYPE` & `object`

## Additional Inherited Members

### 6.33.1 Detailed Description

```
template<typename T>  
class Rcpp::sugar::Any< false, T >
```

Definition at line 58 of file any.h.

### 6.33.2 Member Typedef Documentation

#### 6.33.2.1 PARENT

```
template<typename T >  
typedef SingleLogicalResult< false , Any<false,T> > Rcpp::sugar::Any< false, T >::PARENT
```

Definition at line 61 of file any.h.

#### 6.33.2.2 VEC\_TYPE

```
template<typename T >  
typedef Rcpp::VectorBase<LGLSXP,false,T> Rcpp::sugar::Any< false, T >::VEC_TYPE
```

Definition at line 60 of file any.h.

### 6.33.3 Constructor & Destructor Documentation

#### 6.33.3.1 Any()

```
template<typename T >  
Rcpp::sugar::Any< false, T >::Any (   
    const VEC\_TYPE & t ) [inline]
```

Definition at line 62 of file any.h.

## 6.33.4 Member Function Documentation

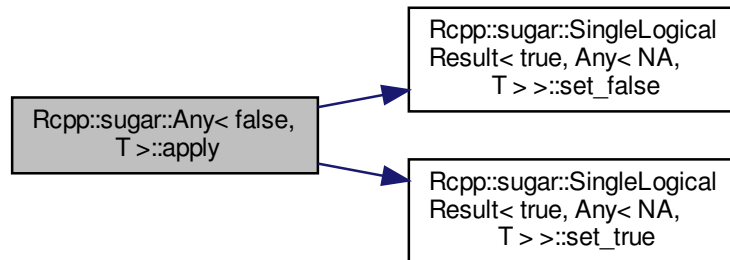
### 6.33.4.1 apply()

```
template<typename T >
void Rcpp::sugar::Any< false, T >::apply ( ) [inline]
```

Definition at line 64 of file any.h.

References Rcpp::sugar::SingleLogicalResult< true, Any< NA, T > >::set\_false(), and Rcpp::sugar::SingleLogicalResult< true, Any< NA, T > >::set\_true().

Here is the call graph for this function:



## 6.33.5 Member Data Documentation

### 6.33.5.1 object

```
template<typename T >
const VEC_TYPE& Rcpp::sugar::Any< false, T >::object [private]
```

Definition at line 75 of file any.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[any.h](#)

## 6.34 Rcpp::Argument Class Reference

```
#include <Named.h>
```

### Public Member Functions

- [Argument](#) ()
- [Argument](#) (const std::string &name\_)
- [template](#)<typename T >  
[traits::named\\_object](#)< T > [operator=](#) (const T &t)

### Public Attributes

- std::string [name](#)

#### 6.34.1 Detailed Description

Definition at line 27 of file Named.h.

#### 6.34.2 Constructor & Destructor Documentation

##### 6.34.2.1 [Argument\(\)](#) [1/2]

```
Rcpp::Argument::Argument ( ) [inline]
```

Definition at line 29 of file Named.h.

##### 6.34.2.2 [Argument\(\)](#) [2/2]

```
Rcpp::Argument::Argument (
    const std::string & name_ ) [inline]
```

Definition at line 30 of file Named.h.

#### 6.34.3 Member Function Documentation

### 6.34.3.1 operator=()

```
template<typename T >
traits::named_object<T> Rcpp::Argument::operator= (
    const T & t ) [inline]
```

Definition at line 33 of file Named.h.

References name.

## 6.34.4 Member Data Documentation

### 6.34.4.1 name

```
std::string Rcpp::Argument::name
```

Definition at line 37 of file Named.h.

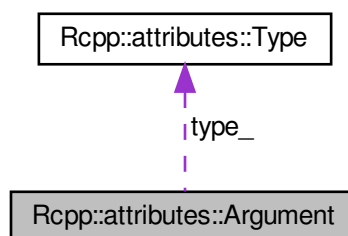
Referenced by operator=().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Named.h](#)

## 6.35 Rcpp::attributes::Argument Class Reference

Collaboration diagram for Rcpp::attributes::Argument:



## Public Member Functions

- [Argument](#) ()
- [Argument](#) (const std::string &name, const [Type](#) &type, const std::string &defaultValue)
- bool [empty](#) () const
- bool [operator==](#) (const [Argument](#) &other) const
- bool [operator!=](#) (const [Argument](#) &other) const
- const std::string & [name](#) () const
- const [Type](#) & [type](#) () const
- const std::string & [defaultValue](#) () const

## Private Attributes

- std::string [name\\_](#)
- [Type](#) [type\\_](#)
- std::string [defaultValue\\_](#)

### 6.35.1 Detailed Description

Definition at line 211 of file attributes.cpp.

### 6.35.2 Constructor & Destructor Documentation

#### 6.35.2.1 [Argument](#)() [1/2]

```
Rcpp::attributes::Argument::Argument ( ) [inline]
```

Definition at line 213 of file attributes.cpp.

#### 6.35.2.2 [Argument](#)() [2/2]

```
Rcpp::attributes::Argument::Argument (
    const std::string & name,
    const Type & type,
    const std::string & defaultValue ) [inline]
```

Definition at line 214 of file attributes.cpp.

### 6.35.3 Member Function Documentation

#### 6.35.3.1 defaultValue()

```
const std::string& Rcpp::attributes::Argument::defaultValue ( ) const [inline]
```

Definition at line 236 of file attributes.cpp.

References `defaultValue_`.

Referenced by `Rcpp::attributes::generateRArgList()`, and `Rcpp::attributes::printArgument()`.

#### 6.35.3.2 empty()

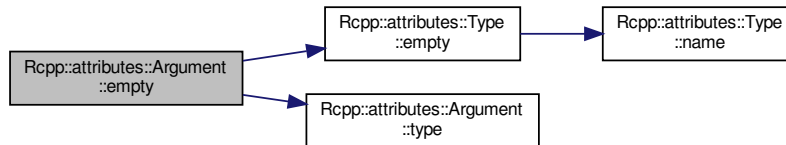
```
bool Rcpp::attributes::Argument::empty ( ) const [inline]
```

Definition at line 221 of file attributes.cpp.

References `Rcpp::attributes::Type::empty()`, and `type()`.

Referenced by `Rcpp::attributes::printArgument()`.

Here is the call graph for this function:



#### 6.35.3.3 name()

```
const std::string& Rcpp::attributes::Argument::name ( ) const [inline]
```

Definition at line 234 of file attributes.cpp.

References `name_`.

Referenced by `Rcpp::attributes::checkRSignature()`, `Rcpp::attributes::generateCpp()`, `Rcpp::attributes::generateRArgList()`, and `Rcpp::attributes::printArgument()`.

#### 6.35.3.4 operator"!=()

```
bool Rcpp::attributes::Argument::operator!= (
    const Argument & other ) const [inline]
```

Definition at line 229 of file attributes.cpp.

#### 6.35.3.5 operator==(())

```
bool Rcpp::attributes::Argument::operator==(
    const Argument & other ) const [inline]
```

Definition at line 223 of file attributes.cpp.

References defaultValue\_, name\_, and type\_.

#### 6.35.3.6 type()

```
const Type& Rcpp::attributes::Argument::type ( ) const [inline]
```

Definition at line 235 of file attributes.cpp.

References type\_.

Referenced by empty(), Rcpp::attributes::generateCpp(), Rcpp::attributes::generateRArgList(), and Rcpp::attributes::printArgument().

### 6.35.4 Member Data Documentation

#### 6.35.4.1 defaultValue\_

```
std::string Rcpp::attributes::Argument::defaultValue_ [private]
```

Definition at line 241 of file attributes.cpp.

Referenced by defaultValue(), and operator==(()).



### 6.35.4.2 name\_

```
std::string Rcpp::attributes::Argument::name_ [private]
```

Definition at line 239 of file attributes.cpp.

Referenced by name(), and operator==().

### 6.35.4.3 type\_

```
Type Rcpp::attributes::Argument::type_ [private]
```

Definition at line 240 of file attributes.cpp.

Referenced by operator==(), and type().

The documentation for this class was generated from the following file:

- [src/attributes.cpp](#)

## 6.36 Rcpp::Armor< T > Class Template Reference

```
#include <Armor.h>
```

### Public Member Functions

- [Armor](#) ()
- [template](#)<typename U >  
[Armor](#) (U x)
- [operator SEXP](#) () const
- [template](#)<typename U >  
[Armor](#) & [operator=](#) (const U &x)
- [~Armor](#) ()
- [template](#)<typename U >  
[Armor](#)< T > & [operator=](#) (const U &x)

### Private Member Functions

- void [init](#) (SEXP x)
- [Armor](#) (const [Armor](#) &)
- [Armor](#) & [operator=](#) (const [Armor](#) &)

## Private Attributes

- [SEXP data](#)
- [PROTECT\\_INDEX index](#)

### 6.36.1 Detailed Description

```
template<typename T>  
class Rcpp::Armor< T >
```

Definition at line 24 of file Armor.h.

### 6.36.2 Constructor & Destructor Documentation

#### 6.36.2.1 Armor() [1/3]

```
template<typename T >  
Rcpp::Armor< T >::Armor ( ) [inline]
```

Definition at line 27 of file Armor.h.

References [Rcpp::Armor< T >::init\(\)](#).

Here is the call graph for this function:



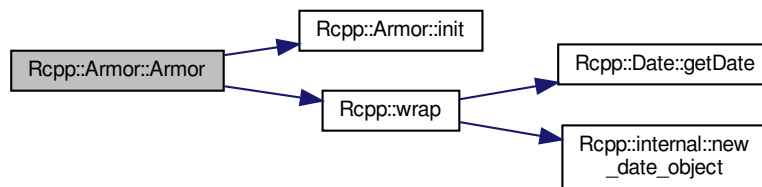
**6.36.2.2 Armor()** [2/3]

```
template<typename T >
template<typename U >
Rcpp::Armor< T >::Armor (
    U x )
```

Definition at line 25 of file protection.h.

References Rcpp::Armor< T >::init(), and Rcpp::wrap().

Here is the call graph for this function:

**6.36.2.3 ~Armor()**

```
template<typename T >
Rcpp::Armor< T >::~~Armor ( ) [inline]
```

Definition at line 40 of file Armor.h.

**6.36.2.4 Armor()** [3/3]

```
template<typename T >
Rcpp::Armor< T >::Armor (
    const Armor< T > & ) [private]
```

**6.36.3 Member Function Documentation**

**6.36.3.1 init()**

```
template<typename T >
void Rcpp::Armor< T >::init (
    SEXP x ) [inline], [private]
```

Definition at line 45 of file Armor.h.

References Rcpp::Armor< T >::data, and Rcpp::Armor< T >::index.

Referenced by Rcpp::Armor< T >::Armor().

**6.36.3.2 operator SEXP()**

```
template<typename T >
Rcpp::Armor< T >::operator SEXP ( ) const [inline]
```

Definition at line 33 of file Armor.h.

References Rcpp::Armor< T >::data.

**6.36.3.3 operator=() [1/3]**

```
template<typename T >
Armor& Rcpp::Armor< T >::operator= (
    const Armor< T > & ) [private]
```

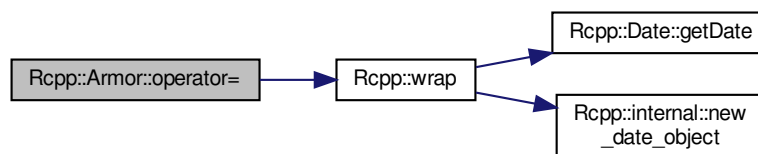
**6.36.3.4 operator=() [2/3]**

```
template<typename T >
template<typename U >
Armor<T>& Rcpp::Armor< T >::operator= (
    const U & x ) [inline]
```

Definition at line 31 of file protection.h.

References Rcpp::wrap().

Here is the call graph for this function:



### 6.36.3.5 operator=() [3/3]

```
template<typename T >
template<typename U >
Armor& Rcpp::Armor< T >::operator= (
    const U & x ) [inline]
```

## 6.36.4 Member Data Documentation

### 6.36.4.1 data

```
template<typename T >
SEXPR Rcpp::Armor< T >::data [private]
```

Definition at line 49 of file Armor.h.

Referenced by Rcpp::Armor< T >::init(), and Rcpp::Armor< T >::operator SEXP().

### 6.36.4.2 index

```
template<typename T >
PROTECT_INDEX Rcpp::Armor< T >::index [private]
```

Definition at line 50 of file Armor.h.

Referenced by Rcpp::Armor< T >::init().

The documentation for this class was generated from the following files:

- [inst/include/Rcpp/protection/Armor.h](#)
- [inst/include/Rcpp/api/meat/protection.h](#)

## 6.37 Rcpp::attributes::Attribute Class Reference

### Public Member Functions

- [Attribute](#) ()
- [Attribute](#) (const std::string &name, const std::vector< [Param](#) > &params, const [Function](#) &function, const std::vector< std::string > &roxygen)
- bool [empty](#) () const
- bool [operator==](#) (const [Attribute](#) &other) const
- bool [operator!=](#) (const [Attribute](#) &other) const
- const std::string & [name](#) () const
- const std::vector< [Param](#) > & [params](#) () const
- [Param](#) [paramNamed](#) (const std::string &name) const
- bool [hasParameter](#) (const std::string &name) const
- const [Function](#) & [function](#) () const
- bool [isExportedFunction](#) () const
- std::string [exportedName](#) () const
- std::string [exportedCppName](#) () const
- bool [rng](#) () const
- bool [invisible](#) () const
- const std::vector< std::string > & [roxygen](#) () const
- std::string [customRSignature](#) () const

### Private Attributes

- std::string [name\\_](#)
- std::vector< [Param](#) > [params\\_](#)
- [Function](#) [function\\_](#)
- std::vector< std::string > [roxygen\\_](#)

#### 6.37.1 Detailed Description

Definition at line 314 of file attributes.cpp.

#### 6.37.2 Constructor & Destructor Documentation

##### 6.37.2.1 Attribute() [1/2]

```
Rcpp::attributes::Attribute::Attribute ( ) [inline]
```

Definition at line 316 of file attributes.cpp.

### 6.37.2.2 Attribute() [2/2]

```
Rcpp::attributes::Attribute::Attribute (
    const std::string & name,
    const std::vector< Param > & params,
    const Function & function,
    const std::vector< std::string > & roxygen ) [inline]
```

Definition at line 317 of file attributes.cpp.

## 6.37.3 Member Function Documentation

### 6.37.3.1 customRSignature()

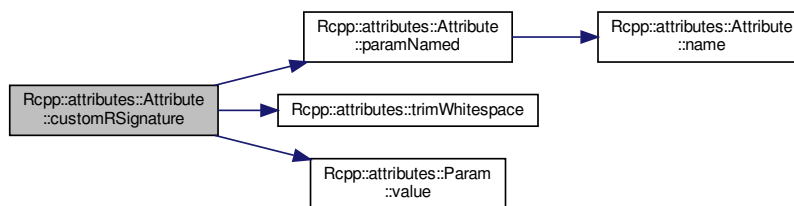
```
std::string Rcpp::attributes::Attribute::customRSignature ( ) const [inline]
```

Definition at line 399 of file attributes.cpp.

References `Rcpp::attributes::kExportSignature`, `paramNamed()`, `Rcpp::attributes::trimWhitespace()`, and `Rcpp::attributes::Param::value()`.

Referenced by `Rcpp::attributes::REExportsGenerator::doWriteFunctions()`.

Here is the call graph for this function:



### 6.37.3.2 empty()

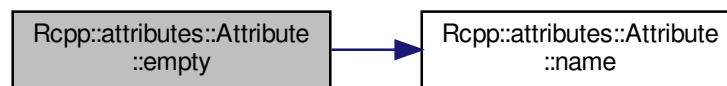
```
bool Rcpp::attributes::Attribute::empty ( ) const [inline]
```

Definition at line 325 of file attributes.cpp.

References name().

Referenced by exportedName(), isExportedFunction(), and Rcpp::attributes::operator<<().

Here is the call graph for this function:



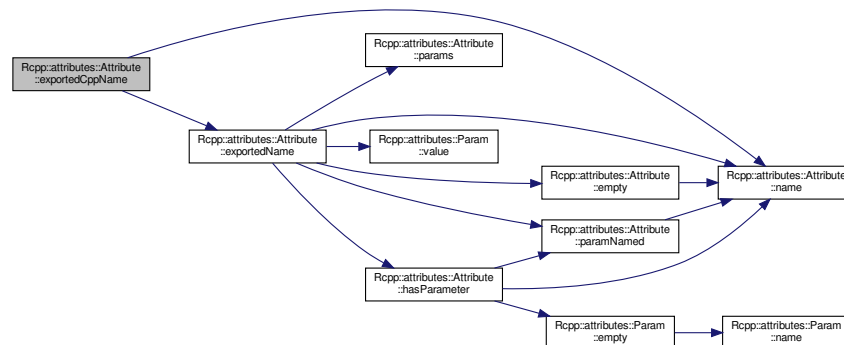
### 6.37.3.3 exportedCppName()

```
std::string Rcpp::attributes::Attribute::exportedCppName ( ) const [inline]
```

Definition at line 373 of file attributes.cpp.

References exportedName(), and name().

Here is the call graph for this function:





### 6.37.3.4 exportedName()

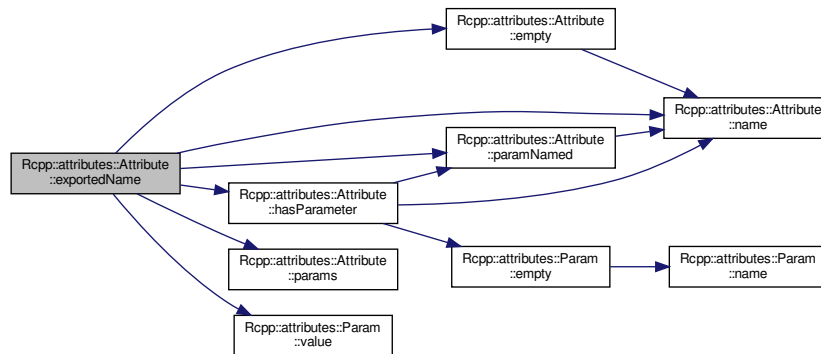
```
std::string Rcpp::attributes::Attribute::exportedName ( ) const [inline]
```

Definition at line 355 of file attributes.cpp.

References `empty()`, `hasParameter()`, `Rcpp::attributes::kExportName`, `name()`, `paramNamed()`, `params()`, and `Rcpp::attributes::Param::value()`.

Referenced by `Rcpp::attributes::REExportsGenerator::doWriteFunctions()`, `exportedCppName()`, and `Rcpp::attributes::CppExportsGenerator::writeEnd()`.

Here is the call graph for this function:



### 6.37.3.5 function()

```
const Function& Rcpp::attributes::Attribute::function ( ) const [inline]
```

Definition at line 349 of file attributes.cpp.

References `function_`.

Referenced by `Rcpp::attributes::REExportsGenerator::doWriteFunctions()`, `Rcpp::attributes::generateCpp()`, `Rcpp::attributes::operator<<()`, and `Rcpp::attributes::CppExportsGenerator::writeEnd()`.

### 6.37.3.6 hasParameter()

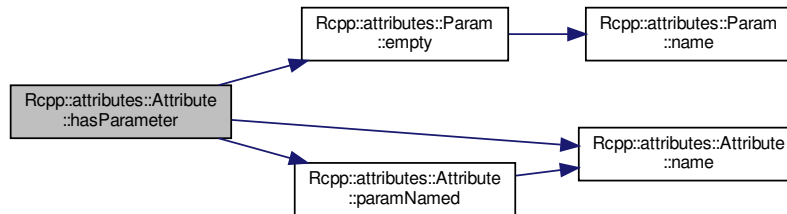
```
bool Rcpp::attributes::Attribute::hasParameter (
    const std::string & name ) const [inline]
```

Definition at line 345 of file attributes.cpp.

References Rcpp::attributes::Param::empty(), name(), and paramNamed().

Referenced by Rcpp::attributes::REExportsGenerator::doWriteFunctions(), and exportedName().

Here is the call graph for this function:



### 6.37.3.7 invisible()

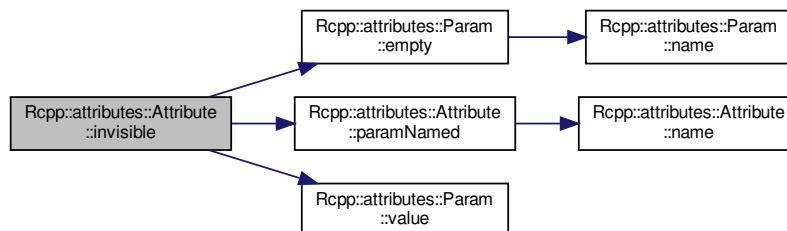
```
bool Rcpp::attributes::Attribute::invisible ( ) const [inline]
```

Definition at line 388 of file attributes.cpp.

References Rcpp::attributes::Param::empty(), Rcpp::attributes::kExportInvisible, Rcpp::attributes::kParamValueTrue, Rcpp::attributes::kParamValueTRUE, paramNamed(), and Rcpp::attributes::Param::value().

Referenced by Rcpp::attributes::REExportsGenerator::doWriteFunctions().

Here is the call graph for this function:



### 6.37.3.8 isExportedFunction()

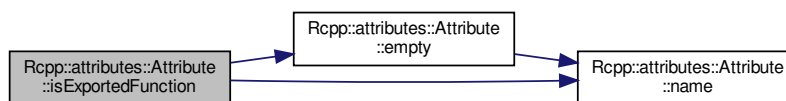
```
bool Rcpp::attributes::Attribute::isExportedFunction ( ) const [inline]
```

Definition at line 351 of file attributes.cpp.

References empty(), Rcpp::attributes::kExportAttribute, and name().

Referenced by Rcpp::attributes::REExportsGenerator::doWriteFunctions(), and Rcpp::attributes::generateCpp().

Here is the call graph for this function:



### 6.37.3.9 name()

```
const std::string& Rcpp::attributes::Attribute::name ( ) const [inline]
```

Definition at line 339 of file attributes.cpp.

References name\_.

Referenced by empty(), exportedCppName(), exportedName(), hasParameter(), isExportedFunction(), Rcpp::attributes::operator<<(), and paramNamed().

### 6.37.3.10 operator"!="()

```
bool Rcpp::attributes::Attribute::operator!= (
    const Attribute & other ) const [inline]
```

Definition at line 334 of file attributes.cpp.

**6.37.3.11 operator==( )**

```
bool Rcpp::attributes::Attribute::operator== (
    const Attribute & other ) const [inline]
```

Definition at line 327 of file attributes.cpp.

References function\_, name\_, params\_, and roxygen\_.

**6.37.3.12 paramNamed( )**

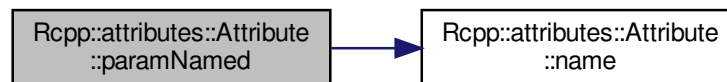
```
Param Rcpp::attributes::Attribute::paramNamed (
    const std::string & name ) const
```

Definition at line 1082 of file attributes.cpp.

References name(), and params\_.

Referenced by customRSignature(), exportedName(), hasParameter(), invisible(), and rng().

Here is the call graph for this function:

**6.37.3.13 params( )**

```
const std::vector<Param>& Rcpp::attributes::Attribute::params ( ) const [inline]
```

Definition at line 341 of file attributes.cpp.

References params\_.

Referenced by exportedName(), and `Rcpp::attributes::operator<<()`.

### 6.37.3.14 rng()

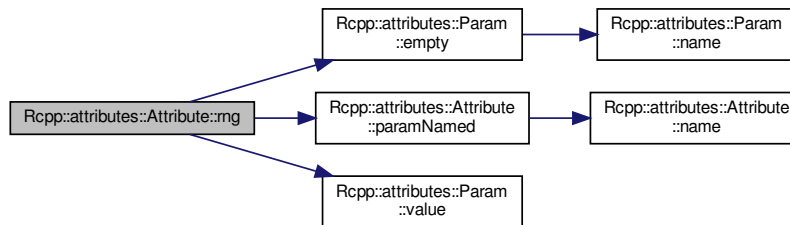
```
bool Rcpp::attributes::Attribute::rng ( ) const [inline]
```

Definition at line 379 of file attributes.cpp.

References `Rcpp::attributes::Param::empty()`, `Rcpp::attributes::kExportRng`, `Rcpp::attributes::kParamValueTrue`, `Rcpp::attributes::kParamValueTRUE`, `paramNamed()`, and `Rcpp::attributes::Param::value()`.

Referenced by `Rcpp::attributes::generateCpp()`.

Here is the call graph for this function:



### 6.37.3.15 roxygen()

```
const std::vector<std::string>& Rcpp::attributes::Attribute::roxygen ( ) const [inline]
```

Definition at line 397 of file attributes.cpp.

References `roxygen_`.

Referenced by `Rcpp::attributes::REExportsGenerator::doWriteFunctions()`.

## 6.37.4 Member Data Documentation

### 6.37.4.1 function\_

```
Function Rcpp::attributes::Attribute::function_ [private]
```

Definition at line 424 of file attributes.cpp.

Referenced by `function()`, and `operator==()`.

#### 6.37.4.2 name\_

```
std::string Rcpp::attributes::Attribute::name_ [private]
```

Definition at line 422 of file attributes.cpp.

Referenced by name(), and operator==().

#### 6.37.4.3 params\_

```
std::vector<Param> Rcpp::attributes::Attribute::params_ [private]
```

Definition at line 423 of file attributes.cpp.

Referenced by operator==(), paramNamed(), and params().

#### 6.37.4.4 roxygen\_

```
std::vector<std::string> Rcpp::attributes::Attribute::roxygen_ [private]
```

Definition at line 425 of file attributes.cpp.

Referenced by operator==(), and roxygen().

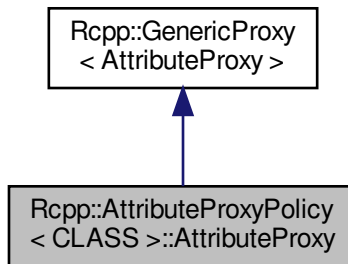
The documentation for this class was generated from the following file:

- [src/attributes.cpp](#)

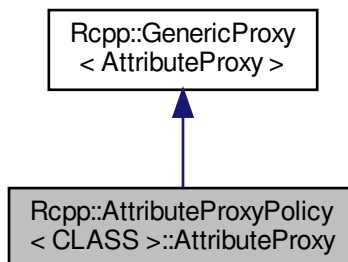
## 6.38 Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy Class Reference

```
#include <AttributeProxy.h>
```

Inheritance diagram for Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy:



Collaboration diagram for Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy:



### Public Member Functions

- [AttributeProxy](#) (CLASS &v, const std::string &name)
- [AttributeProxy](#) & [operator=](#) (const [AttributeProxy](#) &rhs)
- template<typename T >  
[AttributeProxy](#) & [operator=](#) (const T &rhs)
- template<typename T >  
[operator T](#) () const
- [operator SEXP](#) () const
- template<typename T >  
[AttributeProxyPolicy](#)< CLASS >::AttributeProxy & [operator=](#) (const T &rhs)

## Private Member Functions

- SEXP [get](#) () const
- void [set](#) (SEXP x)

## Private Attributes

- CLASS & [parent](#)
- SEXP [attr\\_name](#)

### 6.38.1 Detailed Description

```
template<typename CLASS>
class Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy
```

Definition at line 27 of file AttributeProxy.h.

### 6.38.2 Constructor & Destructor Documentation

#### 6.38.2.1 AttributeProxy()

```
template<typename CLASS >
Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::AttributeProxy (
    CLASS & v,
    const std::string & name ) [inline]
```

Definition at line 29 of file AttributeProxy.h.

### 6.38.3 Member Function Documentation

#### 6.38.3.1 get()

```
template<typename CLASS >
SEXP Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::get ( ) const [inline], [private]
```

Definition at line 48 of file AttributeProxy.h.

References [Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::attr\\_name](#), and [Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::parent](#).

Referenced by [Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator=\(\)](#).



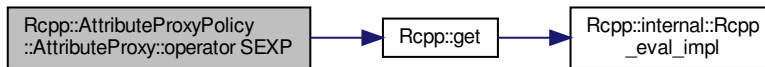
### 6.38.3.2 operator SEXP()

```
template<typename CLASS >  
Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator SEXP [inline]
```

Definition at line 44 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



### 6.38.3.3 operator T()

```
template<typename CLASS >  
template<typename T >  
Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator T
```

Definition at line 39 of file proxy.h.

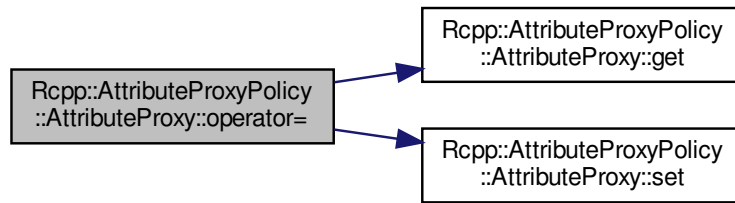
### 6.38.3.4 operator=() [1/3]

```
template<typename CLASS >  
AttributeProxy& Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator= (  
    const AttributeProxy & rhs ) [inline]
```

Definition at line 33 of file AttributeProxy.h.

References Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::get(), and Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::set().

Here is the call graph for this function:



### 6.38.3.5 operator=() [2/3]

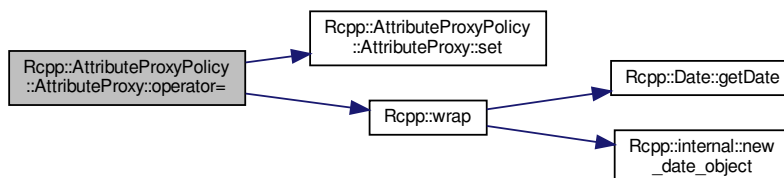
```

template<typename CLASS >
template<typename T >
AttributeProxyPolicy<CLASS>::AttributeProxy & Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator= (
    const T & rhs )
  
```

Definition at line 32 of file proxy.h.

References `Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::set()`, and `Rcpp::wrap()`.

Here is the call graph for this function:



### 6.38.3.6 operator=() [3/3]

```
template<typename CLASS >
template<typename T >
AttributeProxy& Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator= (
    const T & rhs )
```

### 6.38.3.7 set()

```
template<typename CLASS >
void Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::set (
    SEXP x ) [inline], [private]
```

Definition at line 51 of file AttributeProxy.h.

References [Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::attr\\_name](#), and [Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::parent](#).

Referenced by [Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::operator=\(\)](#).

## 6.38.4 Member Data Documentation

### 6.38.4.1 attr\_name

```
template<typename CLASS >
SEXP Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::attr_name [private]
```

Definition at line 46 of file AttributeProxy.h.

Referenced by [Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::get\(\)](#), and [Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::set\(\)](#).

### 6.38.4.2 parent

```
template<typename CLASS >
CLASS& Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::parent [private]
```

Definition at line 45 of file AttributeProxy.h.

Referenced by [Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::get\(\)](#), and [Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy::set\(\)](#).

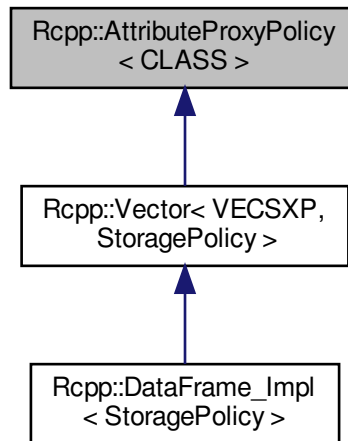
The documentation for this class was generated from the following files:

- [inst/include/Rcpp/proxy/AttributeProxy.h](#)
- [inst/include/Rcpp/api/meat/proxy.h](#)

## 6.39 Rcpp::AttributeProxyPolicy< CLASS > Class Template Reference

```
#include <AttributeProxy.h>
```

Inheritance diagram for Rcpp::AttributeProxyPolicy< CLASS >:



### Classes

- class [AttributeProxy](#)
- class [const\\_AttributeProxy](#)

### Public Member Functions

- [AttributeProxy attr](#) (const std::string &name)
- [const\\_AttributeProxy attr](#) (const std::string &name) const
- std::vector< std::string > [attributeNames](#) () const
- bool [hasAttribute](#) (const std::string &attr) const

### 6.39.1 Detailed Description

```
template<typename CLASS>
class Rcpp::AttributeProxyPolicy< CLASS >
```

Definition at line 24 of file `AttributeProxy.h`.

## 6.39.2 Member Function Documentation

### 6.39.2.1 attr() [1/2]

```
template<typename CLASS >
AttributeProxy Rcpp::AttributeProxyPolicy< CLASS >::attr (
    const std::string & name ) [inline]
```

Definition at line 73 of file AttributeProxy.h.

Referenced by Rcpp::exception::copy\_stack\_trace\_to\_r(), Rcpp::Vector< RTYPE, StoragePolicy >::erase\_range\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::erase\_single\_\_impl(), Rcpp::DataFrame\_Impl< StoragePolicy >::from\_list(), Rcpp::AttributeProxyPolicy< CLASS >::hasAttribute(), Rcpp::Vector< RTYPE, StoragePolicy >::insert\_\_impl(), Rcpp::sugar::na\_omit\_\_impl(), Rcpp::Timer::operator SEXP(), Rcpp::newDatetimeVector::operator=(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_back\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_back\_name\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_front\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_front\_name\_\_impl(), Rcpp::internal::range\_wrap\_dispatch\_\_impl\_\_pair(), and Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

### 6.39.2.2 attr() [2/2]

```
template<typename CLASS >
const AttributeProxy Rcpp::AttributeProxyPolicy< CLASS >::attr (
    const std::string & name ) const [inline]
```

Definition at line 76 of file AttributeProxy.h.

### 6.39.2.3 attributeNames()

```
template<typename CLASS >
std::vector<std::string> Rcpp::AttributeProxyPolicy< CLASS >::attributeNames ( ) const [inline]
```

Definition at line 80 of file AttributeProxy.h.

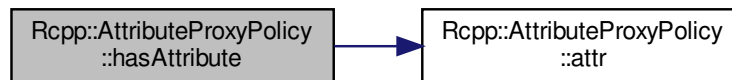
### 6.39.2.4 hasAttribute()

```
template<typename CLASS >
bool Rcpp::AttributeProxyPolicy< CLASS >::hasAttribute (
    const std::string & attr ) const [inline]
```

Definition at line 90 of file AttributeProxy.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr().

Here is the call graph for this function:



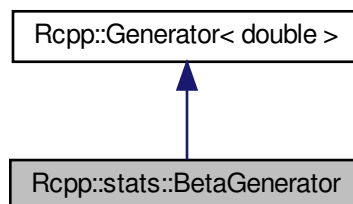
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/proxy/AttributeProxy.h`

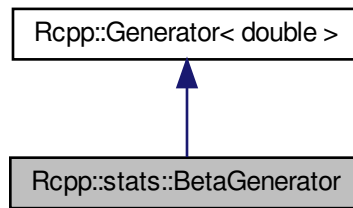
## 6.40 Rcpp::stats::BetaGenerator Class Reference

```
#include <rbeta.h>
```

Inheritance diagram for Rcpp::stats::BetaGenerator:



Collaboration diagram for Rcpp::stats::BetaGenerator:



## Public Member Functions

- [BetaGenerator](#) (double a\_, double b\_)
- double [operator\(\)](#) () const

## Private Attributes

- double [a](#)
- double [b](#)

## Additional Inherited Members

### 6.40.1 Detailed Description

Definition at line 28 of file rbeta.h.

### 6.40.2 Constructor & Destructor Documentation

#### 6.40.2.1 BetaGenerator()

```
Rcpp::stats::BetaGenerator::BetaGenerator (  
    double a_,  
    double b_ ) [inline]
```

Definition at line 30 of file rbeta.h.

### 6.40.3 Member Function Documentation

#### 6.40.3.1 operator()

```
double Rcpp::stats::BetaGenerator::operator() ( ) const [inline]
```

Definition at line 32 of file rbeta.h.

References a, and b.

### 6.40.4 Member Data Documentation

#### 6.40.4.1 a

```
double Rcpp::stats::BetaGenerator::a [private]
```

Definition at line 36 of file rbeta.h.

Referenced by operator()).

#### 6.40.4.2 b

```
double Rcpp::stats::BetaGenerator::b [private]
```

Definition at line 36 of file rbeta.h.

Referenced by operator()).

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rbeta.h](#)

## 6.41 Rcpp::traits::is\_convertible< T, U >::Big Struct Reference

### Public Attributes

- char [dummy](#) [2]



### 6.41.1 Detailed Description

```
template<typename T, typename U>
struct Rcpp::traits::is_convertible< T, U >::Big
```

Definition at line 32 of file is\_convertible.h.

### 6.41.2 Member Data Documentation

#### 6.41.2.1 dummy

```
template<typename T , typename U >
char Rcpp::traits::is_convertible< T, U >::Big::dummy[2]
```

Definition at line 32 of file is\_convertible.h.

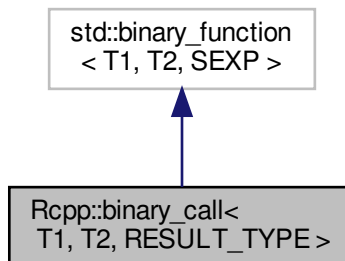
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_convertible.h](#)

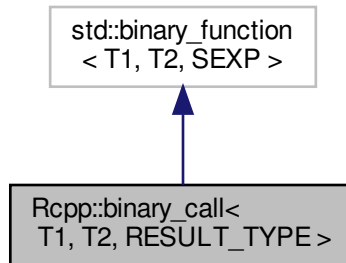
## 6.42 Rcpp::binary\_call< T1, T2, RESULT\_TYPE > Class Template Reference

```
#include <Language.h>
```

Inheritance diagram for Rcpp::binary\_call< T1, T2, RESULT\_TYPE >:



Collaboration diagram for `Rcpp::binary_call< T1, T2, RESULT_TYPE >`:



## Public Member Functions

- `binary_call` ([Language](#) call\_)
- `binary_call` ([Language](#) call\_, [R\\_xlen\\_t](#) index1, [R\\_xlen\\_t](#) index2)
- `binary_call` ([Function](#) fun)
- `RESULT_TYPE operator()` (const T1 &o1, const T2 &o2)

## Private Attributes

- [Language](#) call
- [Language::Proxy](#) proxy1
- [Language::Proxy](#) proxy2

### 6.42.1 Detailed Description

```
template<typename T1, typename T2, typename RESULT_TYPE = SEXP>
class Rcpp::binary_call< T1, T2, RESULT_TYPE >
```

Definition at line 201 of file `Language.h`.

### 6.42.2 Constructor & Destructor Documentation

### 6.42.2.1 binary\_call() [1/3]

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>
Rcpp::binary_call< T1, T2, RESULT_TYPE >::binary_call (
    Language call_ ) [inline]
```

Definition at line 206 of file Language.h.

### 6.42.2.2 binary\_call() [2/3]

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>
Rcpp::binary_call< T1, T2, RESULT_TYPE >::binary_call (
    Language call_,
    R_xlen_t index1,
    R_xlen_t index2 ) [inline]
```

Definition at line 207 of file Language.h.

### 6.42.2.3 binary\_call() [3/3]

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>
Rcpp::binary_call< T1, T2, RESULT_TYPE >::binary_call (
    Function fun ) [inline]
```

Definition at line 208 of file Language.h.

## 6.42.3 Member Function Documentation

### 6.42.3.1 operator>()

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>
RESULT_TYPE Rcpp::binary_call< T1, T2, RESULT_TYPE >::operator() (
    const T1 & o1,
    const T2 & o2 ) [inline]
```

Definition at line 210 of file Language.h.

References Rcpp::unary\_call< T, RESULT\_TYPE >::call.

## 6.42.4 Member Data Documentation

### 6.42.4.1 call

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>  
Language Rcpp::binary_call< T1, T2, RESULT_TYPE >::call [private]
```

Definition at line 217 of file Language.h.

### 6.42.4.2 proxy1

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>  
Language::Proxy Rcpp::binary_call< T1, T2, RESULT_TYPE >::proxy1 [private]
```

Definition at line 218 of file Language.h.

### 6.42.4.3 proxy2

```
template<typename T1 , typename T2 , typename RESULT_TYPE = SEXP>  
Language::Proxy Rcpp::binary_call< T1, T2, RESULT_TYPE >::proxy2 [private]
```

Definition at line 219 of file Language.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Language.h](#)

## 6.43 Rcpp::sugar::cbind\_impl::BindableExpression< RTYPE, E > Class Template Reference

```
#include <cbind.h>
```

### Public Types

- typedef [cbind\\_storage\\_type](#)< RTYPE >::type [stored\\_type](#)

## Public Member Functions

- [stored\\_type operator\[\]](#) (R\_xlen\_t i) const
- [stored\\_type operator\(\)](#) (R\_xlen\_t i, R\_xlen\_t j) const
- [R\\_xlen\\_t size](#) () const
- [R\\_xlen\\_t nrow](#) () const
- [R\\_xlen\\_t ncol](#) () const
- [operator E&](#) ()
- [operator const E &](#) () const

### 6.43.1 Detailed Description

```
template<int RTYPE, typename E>
class Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >
```

Definition at line 52 of file cbind.h.

### 6.43.2 Member Typedef Documentation

#### 6.43.2.1 stored\_type

```
template<int RTYPE, typename E >
typedef cbind\_storage\_type<RTYPE>::type Rcpp::sugar::cbind\_impl::BindableExpression< RTYPE, E
>::stored\_type
```

Definition at line 54 of file cbind.h.

### 6.43.3 Member Function Documentation

#### 6.43.3.1 ncol()

```
template<int RTYPE, typename E >
R_xlen_t Rcpp::sugar::cbind\_impl::BindableExpression< RTYPE, E >::ncol ( ) const [inline]
```

Definition at line 72 of file cbind.h.

### 6.43.3.2 nrow()

```
template<int RTYPE, typename E >
R_xlen_t Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::nrow ( ) const [inline]
```

Definition at line 68 of file cbind.h.

### 6.43.3.3 operator const E &()

```
template<int RTYPE, typename E >
Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::operator const E & ( ) const [inline]
```

Definition at line 77 of file cbind.h.

### 6.43.3.4 operator E&()

```
template<int RTYPE, typename E >
Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::operator E& ( ) [inline]
```

Definition at line 76 of file cbind.h.

### 6.43.3.5 operator>()()

```
template<int RTYPE, typename E >
stored_type Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::operator() (
    R_xlen_t i,
    R_xlen_t j ) const [inline]
```

Definition at line 60 of file cbind.h.

### 6.43.3.6 operator[]()

```
template<int RTYPE, typename E >
stored_type Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 56 of file cbind.h.

### 6.43.3.7 size()

```
template<int RTYPE, typename E >  
R_xlen_t Rcpp::sugar::cbind_impl::BindableExpression< RTYPE, E >::size ( ) const [inline]
```

Definition at line 64 of file cbind.h.

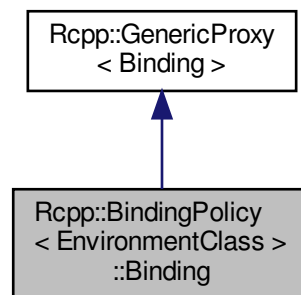
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

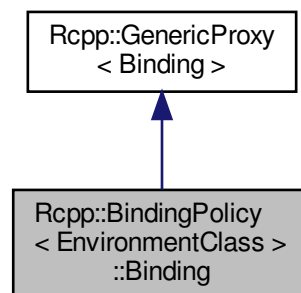
## 6.44 Rcpp::BindingPolicy< EnvironmentClass >::Binding Class Reference

```
#include <Binding.h>
```

Inheritance diagram for Rcpp::BindingPolicy< EnvironmentClass >::Binding:



Collaboration diagram for Rcpp::BindingPolicy< EnvironmentClass >::Binding:



## Public Member Functions

- [Binding](#) (EnvironmentClass &env\_, const std::string &name\_)
- bool [active](#) () const
- bool [locked](#) () const
- bool [exists](#) () const
- void [lock](#) ()
- void [unlock](#) ()
- [Binding](#) & [operator=](#) (const [Binding](#) &rhs)
- template<typename WRAPPABLE >  
[Binding](#) & [operator=](#) (const WRAPPABLE &rhs)
- template<typename T >  
[operator T](#) () const
- template<typename T >  
[BindingPolicy](#)< CLASS >::[Binding](#) & [operator=](#) (const T &rhs)

## Private Member Functions

- SEXP [get](#) () const
- void [set](#) (SEXP x)

## Private Attributes

- EnvironmentClass & [env](#)
- std::string [name](#)

### 6.44.1 Detailed Description

```
template<typename EnvironmentClass>
class Rcpp::BindingPolicy< EnvironmentClass >::Binding
```

Definition at line 27 of file Binding.h.

### 6.44.2 Constructor & Destructor Documentation

#### 6.44.2.1 Binding()

```
template<typename EnvironmentClass >
Rcpp::BindingPolicy< EnvironmentClass >::Binding::Binding (
    EnvironmentClass & env_,
    const std::string & name_ ) [inline]
```

Definition at line 29 of file Binding.h.



### 6.44.3 Member Function Documentation

#### 6.44.3.1 active()

```
template<typename EnvironmentClass >  
bool Rcpp::BindingPolicy< EnvironmentClass >::Binding::active ( ) const [inline]
```

Definition at line 32 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

#### 6.44.3.2 exists()

```
template<typename EnvironmentClass >  
bool Rcpp::BindingPolicy< EnvironmentClass >::Binding::exists ( ) const [inline]
```

Definition at line 38 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

#### 6.44.3.3 get()

```
template<typename EnvironmentClass >  
SEXPR Rcpp::BindingPolicy< EnvironmentClass >::Binding::get ( ) const [inline], [private]
```

Definition at line 60 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

Referenced by Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator=().

#### 6.44.3.4 lock()

```
template<typename EnvironmentClass >
void Rcpp::BindingPolicy< EnvironmentClass >::Binding::lock ( ) [inline]
```

Definition at line 41 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

#### 6.44.3.5 locked()

```
template<typename EnvironmentClass >
bool Rcpp::BindingPolicy< EnvironmentClass >::Binding::locked ( ) const [inline]
```

Definition at line 35 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

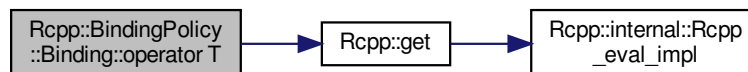
#### 6.44.3.6 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::BindingPolicy< CLASS >::Binding::operator T
```

Definition at line 137 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



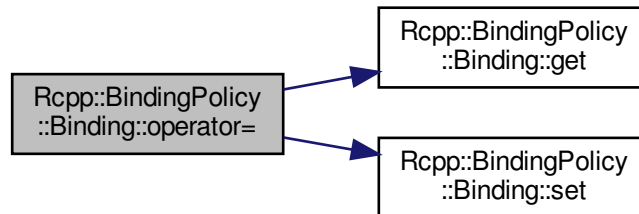
**6.44.3.7 operator=()** [1/3]

```
template<typename EnvironmentClass >
Binding& Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator= (
    const Binding & rhs ) [inline]
```

Definition at line 47 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::get(), and Rcpp::BindingPolicy< EnvironmentClass >::Binding::set().

Here is the call graph for this function:

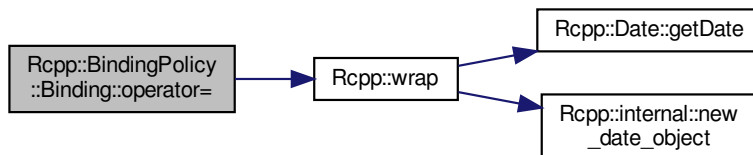
**6.44.3.8 operator=()** [2/3]

```
template<typename EnvironmentClass >
template<typename T >
BindingPolicy<CLASS>::Binding& Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator= (
    const T & rhs )
```

Definition at line 130 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:



### 6.44.3.9 operator=() [3/3]

```
template<typename EnvironmentClass >
template<typename WRAPPABLE >
Binding& Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator= (
    const WRAPPABLE & rhs )
```

### 6.44.3.10 set()

```
template<typename EnvironmentClass >
void Rcpp::BindingPolicy< EnvironmentClass >::Binding::set (
    SEXP x ) [inline], [private]
```

Definition at line 64 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

Referenced by Rcpp::BindingPolicy< EnvironmentClass >::Binding::operator=().

### 6.44.3.11 unlock()

```
template<typename EnvironmentClass >
void Rcpp::BindingPolicy< EnvironmentClass >::Binding::unlock ( ) [inline]
```

Definition at line 44 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::Binding::name.

## 6.44.4 Member Data Documentation

### 6.44.4.1 env

```
template<typename EnvironmentClass >
EnvironmentClass& Rcpp::BindingPolicy< EnvironmentClass >::Binding::env [private]
```

Definition at line 68 of file Binding.h.

Referenced by Rcpp::BindingPolicy< EnvironmentClass >::Binding::active(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::exists(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::get(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::lock(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::locked(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::set(), and Rcpp::BindingPolicy< EnvironmentClass >::Binding::unlock().

#### 6.44.4.2 name

```
template<typename EnvironmentClass >
std::string Rcpp::BindingPolicy< EnvironmentClass >::Binding::name [private]
```

Definition at line 69 of file Binding.h.

Referenced by Rcpp::BindingPolicy< EnvironmentClass >::Binding::active(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::exists(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::get(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::lock(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::locked(), Rcpp::BindingPolicy< EnvironmentClass >::Binding::set(), and Rcpp::BindingPolicy< EnvironmentClass >::Binding::unlock().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/Binding.h
- inst/include/Rcpp/api/meat/proxy.h

## 6.45 Rcpp::BindingPolicy< EnvironmentClass > Class Template Reference

```
#include <Binding.h>
```

### Classes

- class [Binding](#)
- class [const\\_Binding](#)

### Public Member Functions

- [const\\_Binding operator\[\]](#) (const std::string &name) const
- [Binding operator\[\]](#) (const std::string &name)

#### 6.45.1 Detailed Description

```
template<typename EnvironmentClass>
class Rcpp::BindingPolicy< EnvironmentClass >
```

Definition at line 24 of file Binding.h.

#### 6.45.2 Member Function Documentation

### 6.45.2.1 operator[]() [1/2]

```
template<typename EnvironmentClass >  
Binding Rcpp::BindingPolicy< EnvironmentClass >::operator[] (   
    const std::string & name ) [inline]
```

Definition at line 101 of file Binding.h.

### 6.45.2.2 operator[]() [2/2]

```
template<typename EnvironmentClass >  
const_Binding Rcpp::BindingPolicy< EnvironmentClass >::operator[] (   
    const std::string & name ) const [inline]
```

Definition at line 98 of file Binding.h.

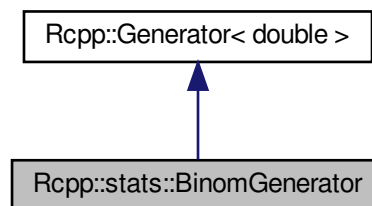
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/Binding.h

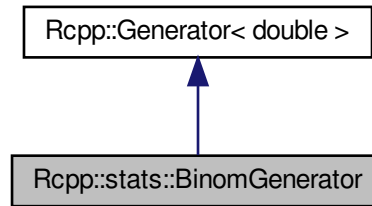
## 6.46 Rcpp::stats::BinomGenerator Class Reference

```
#include <rbinom.h>
```

Inheritance diagram for Rcpp::stats::BinomGenerator:



Collaboration diagram for Rcpp::stats::BinomGenerator:



## Public Member Functions

- `BinomGenerator` (double `nin_`, double `pp_`)
- double `operator()` () const

## Private Attributes

- double `nin`
- double `pp`

## Additional Inherited Members

### 6.46.1 Detailed Description

Definition at line 28 of file `rbinom.h`.

### 6.46.2 Constructor & Destructor Documentation

#### 6.46.2.1 BinomGenerator()

```
Rcpp::stats::BinomGenerator::BinomGenerator (  
    double nin_,  
    double pp_ ) [inline]
```

Definition at line 30 of file `rbinom.h`.

### 6.46.3 Member Function Documentation

#### 6.46.3.1 operator()

```
double Rcpp::stats::BinomGenerator::operator() ( ) const [inline]
```

Definition at line 31 of file rbinom.h.

References [nin](#), and [pp](#).

### 6.46.4 Member Data Documentation

#### 6.46.4.1 nin

```
double Rcpp::stats::BinomGenerator::nin [private]
```

Definition at line 35 of file rbinom.h.

Referenced by [operator\(\)](#).

#### 6.46.4.2 pp

```
double Rcpp::stats::BinomGenerator::pp [private]
```

Definition at line 35 of file rbinom.h.

Referenced by [operator\(\)](#).

The documentation for this class was generated from the following file:

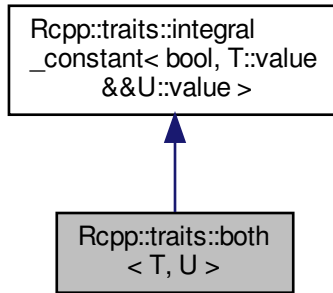
- [inst/include/Rcpp/stats/random/rbinom.h](#)



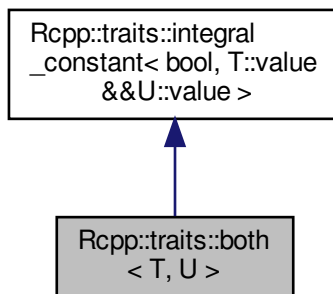
## 6.47 Rcpp::traits::both< T, U > Struct Template Reference

```
#include <integral_constant.h>
```

Inheritance diagram for Rcpp::traits::both< T, U >:



Collaboration diagram for Rcpp::traits::both< T, U >:



### Additional Inherited Members

#### 6.47.1 Detailed Description

```
template<typename T, typename U>  
struct Rcpp::traits::both< T, U >
```

Definition at line 37 of file integral\_constant.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/integral\\_constant.h](#)

## 6.48 Cache Class Reference

```
#include <convolve10_cpp.h>
```

### Public Types

- typedef double & [proxy](#)
- typedef double \* [iterator](#)
- typedef double & [proxy](#)
- typedef double \* [iterator](#)

### Public Member Functions

- [Cache](#) (iterator data\_)
- [proxy ref](#) (int i)
- [proxy ref](#) (int i) const
- [Cache](#) (iterator data\_)
- [proxy ref](#) (int i)
- [proxy ref](#) (int i) const

### Private Attributes

- [iterator data](#)

#### 6.48.1 Detailed Description

Definition at line 2 of file convolve10\_cpp.h.

#### 6.48.2 Member Typedef Documentation

##### 6.48.2.1 iterator [1/2]

```
typedef double* Cache::iterator
```

Definition at line 5 of file convolve10\_cpp.h.

### 6.48.2.2 iterator [2/2]

```
typedef double* Cache::iterator
```

Definition at line 18 of file convolve9\_cpp.cpp.

### 6.48.2.3 proxy [1/2]

```
typedef double& Cache::proxy
```

Definition at line 4 of file convolve10\_cpp.h.

### 6.48.2.4 proxy [2/2]

```
typedef double& Cache::proxy
```

Definition at line 17 of file convolve9\_cpp.cpp.

## 6.48.3 Constructor & Destructor Documentation

### 6.48.3.1 Cache() [1/2]

```
Cache::Cache (  
    iterator data_ ) [inline]
```

Definition at line 7 of file convolve10\_cpp.h.

### 6.48.3.2 Cache() [2/2]

```
Cache::Cache (  
    iterator data_ ) [inline]
```

Definition at line 20 of file convolve9\_cpp.cpp.

## 6.48.4 Member Function Documentation

### 6.48.4.1 ref() [1/4]

```
proxy Cache::ref (  
    int i ) [inline]
```

Definition at line 9 of file convolve10\_cpp.h.

References data.

Referenced by Vec::operator[]().

### 6.48.4.2 ref() [2/4]

```
proxy Cache::ref (  
    int i ) [inline]
```

Definition at line 22 of file convolve9\_cpp.cpp.

References data.

### 6.48.4.3 ref() [3/4]

```
proxy Cache::ref (  
    int i ) const [inline]
```

Definition at line 10 of file convolve10\_cpp.h.

References data.

### 6.48.4.4 ref() [4/4]

```
proxy Cache::ref (  
    int i ) const [inline]
```

Definition at line 23 of file convolve9\_cpp.cpp.

References data.

## 6.48.5 Member Data Documentation

### 6.48.5.1 data

`iterator` Cache::data [private]

Definition at line 13 of file convolve10\_cpp.h.

Referenced by ref().

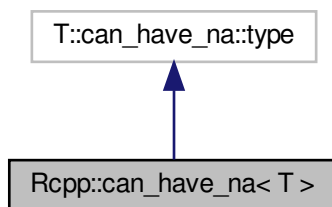
The documentation for this class was generated from the following files:

- inst/examples/ConvolveBenchmarks/convolve10\_cpp.h
- inst/examples/ConvolveBenchmarks/convolve9\_cpp.cpp

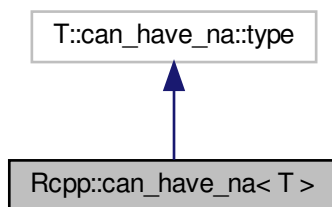
## 6.49 Rcpp::can\_have\_na< T > Struct Template Reference

```
#include <can_have_na.h>
```

Inheritance diagram for Rcpp::can\_have\_na< T >:



Collaboration diagram for Rcpp::can\_have\_na< T >:



### 6.49.1 Detailed Description

```
template<typename T>
struct Rcpp::can_have_na< T >
```

Definition at line 28 of file can\_have\_na.h.

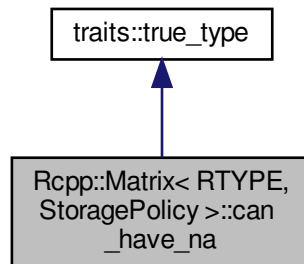
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/logical/can\_have\_na.h

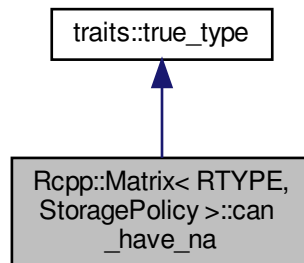
## 6.50 Rcpp::Matrix< RTYPE, StoragePolicy >::can\_have\_na Struct Reference

```
#include <Matrix.h>
```

Inheritance diagram for Rcpp::Matrix< RTYPE, StoragePolicy >::can\_have\_na:



Collaboration diagram for Rcpp::Matrix< RTYPE, StoragePolicy >::can\_have\_na:



## Additional Inherited Members

### 6.50.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
struct Rcpp::Matrix< RTYPE, StoragePolicy >::can_have_na
```

Definition at line 35 of file Matrix.h.

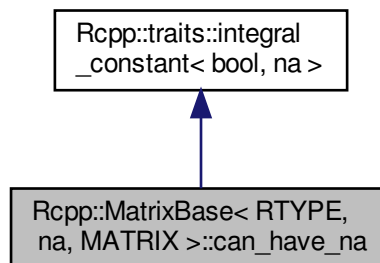
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/Matrix.h](#)

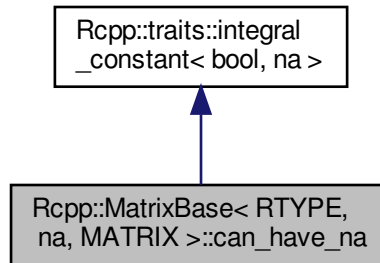
## 6.51 Rcpp::MatrixBase< RTYPE, na, MATRIX >::can\_have\_na Struct Reference

```
#include <MatrixBase.h>
```

Inheritance diagram for Rcpp::MatrixBase< RTYPE, na, MATRIX >::can\_have\_na:



Collaboration diagram for `Rcpp::MatrixBase< RTYPE, na, MATRIX >::can_have_na`:



## Additional Inherited Members

### 6.51.1 Detailed Description

```

template<int RTYPE, bool na, typename MATRIX>
struct Rcpp::MatrixBase< RTYPE, na, MATRIX >::can_have_na
  
```

Definition at line 36 of file `MatrixBase.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/vector/MatrixBase.h`

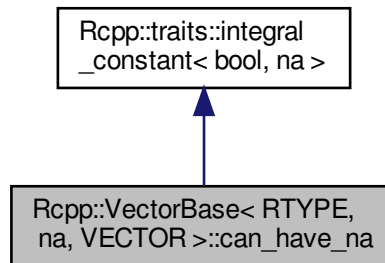
## 6.52 `Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na` Struct Reference

```

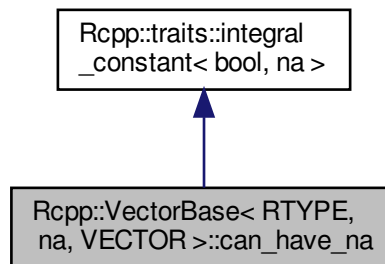
#include <VectorBase.h>
  
```



Inheritance diagram for Rcpp::VectorBase< RTYPE, na, VECTOR >::can\_have\_na:



Collaboration diagram for Rcpp::VectorBase< RTYPE, na, VECTOR >::can\_have\_na:



## Additional Inherited Members

### 6.52.1 Detailed Description

```

template<int RTYPE, bool na, typename VECTOR>
struct Rcpp::VectorBase< RTYPE, na, VECTOR >::can_have_na
  
```

Definition at line 33 of file VectorBase.h.

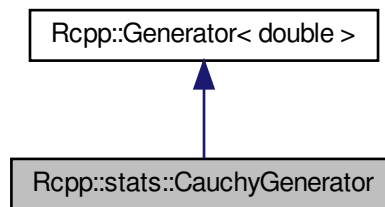
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/[VectorBase.h](#)

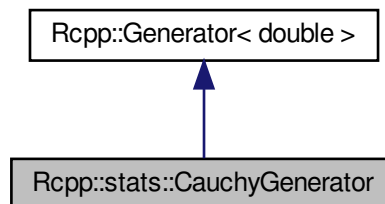
## 6.53 Rcpp::stats::CauchyGenerator Class Reference

```
#include <rcauchy.h>
```

Inheritance diagram for Rcpp::stats::CauchyGenerator:



Collaboration diagram for Rcpp::stats::CauchyGenerator:



### Public Member Functions

- [CauchyGenerator](#) (double location\_, double scale\_)
- double [operator\(\)](#) () const

### Private Attributes

- double [location](#)
- double [scale](#)

## Additional Inherited Members

### 6.53.1 Detailed Description

Definition at line 28 of file rcauchy.h.

### 6.53.2 Constructor & Destructor Documentation

#### 6.53.2.1 CauchyGenerator()

```
Rcpp::stats::CauchyGenerator::CauchyGenerator (
    double location_,
    double scale_ ) [inline]
```

Definition at line 31 of file rcauchy.h.

### 6.53.3 Member Function Documentation

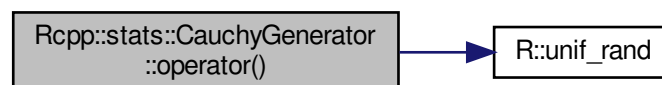
#### 6.53.3.1 operator>()

```
double Rcpp::stats::CauchyGenerator::operator() ( ) const [inline]
```

Definition at line 34 of file rcauchy.h.

References location, scale, and R::unif\_rand().

Here is the call graph for this function:



## 6.53.4 Member Data Documentation

### 6.53.4.1 location

```
double Rcpp::stats::CauchyGenerator::location [private]
```

Definition at line 39 of file `rcauchy.h`.

Referenced by `operator()`.

### 6.53.4.2 scale

```
double Rcpp::stats::CauchyGenerator::scale [private]
```

Definition at line 39 of file `rcauchy.h`.

Referenced by `operator()`.

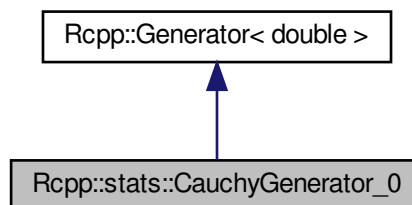
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/stats/random/rcauchy.h`

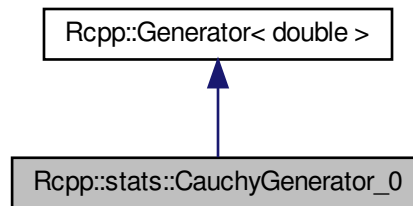
## 6.54 Rcpp::stats::CauchyGenerator\_0 Class Reference

```
#include <rcauchy.h>
```

Inheritance diagram for `Rcpp::stats::CauchyGenerator_0`:



Collaboration diagram for Rcpp::stats::CauchyGenerator\_0:



## Public Member Functions

- [CauchyGenerator\\_0\(\)](#)
- double [operator\(\)](#) () const

## Additional Inherited Members

### 6.54.1 Detailed Description

Definition at line 56 of file rcauchy.h.

### 6.54.2 Constructor & Destructor Documentation

#### 6.54.2.1 CauchyGenerator\_0()

```
Rcpp::stats::CauchyGenerator_0::CauchyGenerator_0 ( ) [inline]
```

Definition at line 59 of file rcauchy.h.

### 6.54.3 Member Function Documentation

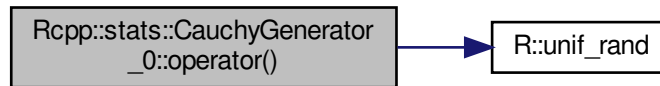
### 6.54.3.1 operator()

```
double Rcpp::stats::CauchyGenerator_0::operator() ( ) const [inline]
```

Definition at line 61 of file rcauchy.h.

References R::unif\_rand().

Here is the call graph for this function:



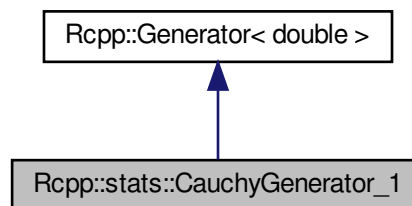
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rcauchy.h](#)

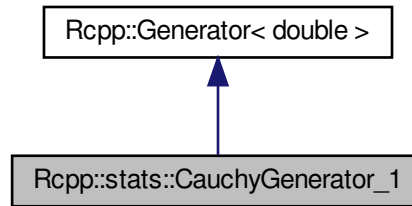
## 6.55 Rcpp::stats::CauchyGenerator\_1 Class Reference

```
#include <rcauchy.h>
```

Inheritance diagram for Rcpp::stats::CauchyGenerator\_1:



Collaboration diagram for Rcpp::stats::CauchyGenerator\_1:



## Public Member Functions

- [CauchyGenerator\\_1](#) (double location\_)
- double [operator\(\)](#) () const

## Private Attributes

- double [location](#)

## Additional Inherited Members

### 6.55.1 Detailed Description

Definition at line 42 of file `rcauchy.h`.

### 6.55.2 Constructor & Destructor Documentation

#### 6.55.2.1 CauchyGenerator\_1()

```
Rcpp::stats::CauchyGenerator_1::CauchyGenerator_1 (  
    double location_ ) [inline]
```

Definition at line 45 of file `rcauchy.h`.

### 6.55.3 Member Function Documentation

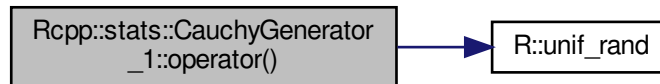
#### 6.55.3.1 operator()

```
double Rcpp::stats::CauchyGenerator_1::operator() ( ) const [inline]
```

Definition at line 48 of file rcauchy.h.

References location, and R::unif\_rand().

Here is the call graph for this function:



### 6.55.4 Member Data Documentation

#### 6.55.4.1 location

```
double Rcpp::stats::CauchyGenerator_1::location [private]
```

Definition at line 53 of file rcauchy.h.

Referenced by operator().

The documentation for this class was generated from the following file:

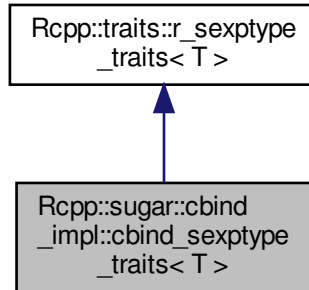
- [inst/include/Rcpp/stats/random/rcauchy.h](#)



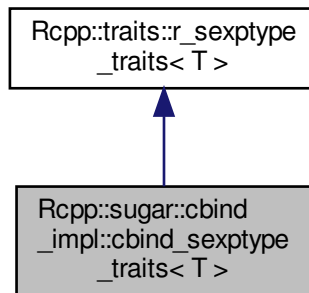
## 6.56 Rcpp::sugar::cbind\_impl::cbind\_sexptype\_traits< T > Struct Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind\_impl::cbind\_sexptype\_traits< T >:



Collaboration diagram for Rcpp::sugar::cbind\_impl::cbind\_sexptype\_traits< T >:



### Additional Inherited Members

#### 6.56.1 Detailed Description

```
template<typename T>
struct Rcpp::sugar::cbind_impl::cbind_sexptype_traits< T >
```

Definition at line 31 of file cbind.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/cbind.h](#)

## 6.57 Rcpp::sugar::cbind\_impl::cbind\_sexptype\_traits< SEXP > Struct Reference

```
#include <cbind.h>
```

### Public Types

- enum { [rtype](#) = STRSXP }

### 6.57.1 Detailed Description

Definition at line 35 of file cbind.h.

### 6.57.2 Member Enumeration Documentation

#### 6.57.2.1 anonymous enum

anonymous enum

Enumerator

rtype	
-------	--

Definition at line 36 of file cbind.h.

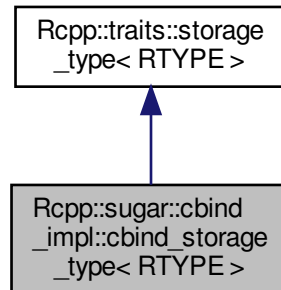
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/cbind.h](#)

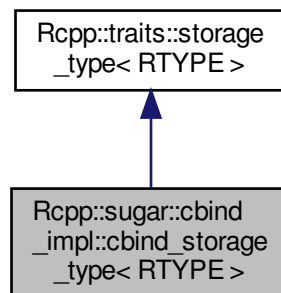
## 6.58 Rcpp::sugar::cbind\_impl::cbind\_storage\_type< RTYPE > Struct Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind\_impl::cbind\_storage\_type< RTYPE >:



Collaboration diagram for Rcpp::sugar::cbind\_impl::cbind\_storage\_type< RTYPE >:



### Additional Inherited Members

#### 6.58.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::sugar::cbind_impl::cbind_storage_type< RTYPE >
```

Definition at line 41 of file cbind.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/cbind.h](#)

## 6.59 Rcpp::sugar::cbind\_impl::cbind\_storage\_type< LGLSXP > Struct Reference

```
#include <cbind.h>
```

### Public Types

- typedef bool [type](#)

#### 6.59.1 Detailed Description

Definition at line 45 of file cbind.h.

#### 6.59.2 Member Typedef Documentation

##### 6.59.2.1 type

```
typedef bool Rcpp::sugar::cbind_impl::cbind_storage_type< LGLSXP >::type
```

Definition at line 46 of file cbind.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/cbind.h](#)

## 6.60 Rcpp::traits::char\_type< T > Struct Template Reference

```
#include <char_type.h>
```

## Public Types

- typedef T::value\_type [type](#)

### 6.60.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::char_type< T >
```

Definition at line 30 of file char\_type.h.

### 6.60.2 Member Typedef Documentation

#### 6.60.2.1 type

```
template<typename T >  
typedef T::value_type Rcpp::traits::char\_type< T >::type
```

Definition at line 31 of file char\_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[char\\_type.h](#)

## 6.61 Rcpp::traits::char\_type< const char \* > Struct Reference

```
#include <char_type.h>
```

## Public Types

- typedef char [type](#)

### 6.61.1 Detailed Description

Definition at line 37 of file char\_type.h.

### 6.61.2 Member Typedef Documentation

### 6.61.2.1 type

```
typedef char Rcpp::traits::char_type< const char * >::type
```

Definition at line 38 of file char\_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[char\\_type.h](#)

## 6.62 Rcpp::traits::char\_type< const wchar\_t \* > Struct Reference

```
#include <char_type.h>
```

### Public Types

- typedef wchar\_t [type](#)

### 6.62.1 Detailed Description

Definition at line 34 of file char\_type.h.

### 6.62.2 Member Typedef Documentation

#### 6.62.2.1 type

```
typedef wchar_t Rcpp::traits::char_type< const wchar_t * >::type
```

Definition at line 35 of file char\_type.h.

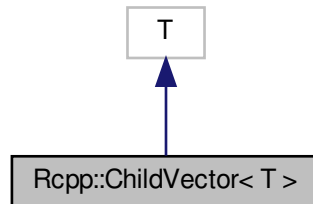
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[char\\_type.h](#)

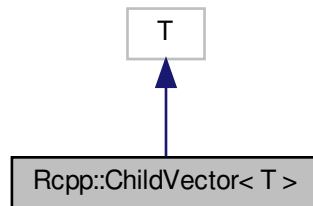
## 6.63 Rcpp::ChildVector< T > Class Template Reference

```
#include <ChildVector.h>
```

Inheritance diagram for Rcpp::ChildVector< T >:



Collaboration diagram for Rcpp::ChildVector< T >:



### Public Member Functions

- [ChildVector](#) (SEXP data\_, SEXP parent\_, R\_xlen\_t i\_)
- [ChildVector](#) (const [ChildVector](#) &other)
- [ChildVector](#) & [operator=](#) (const [ChildVector](#) &other)
- [ChildVector](#) & [operator=](#) (const T &other)
- `template<typename U >`  
[ChildVector](#) & [operator=](#) (const U &other)

### Private Attributes

- SEXP [parent](#)
- R\_xlen\_t [i](#)

### 6.63.1 Detailed Description

```
template<typename T>  
class Rcpp::ChildVector< T >
```

Definition at line 26 of file ChildVector.h.

### 6.63.2 Constructor & Destructor Documentation

#### 6.63.2.1 ChildVector() [1/2]

```
template<typename T >  
Rcpp::ChildVector< T >::ChildVector (   
    SEXP data_,  
    SEXP parent_,  
    R_xlen_t i_ ) [inline]
```

Definition at line 30 of file ChildVector.h.

#### 6.63.2.2 ChildVector() [2/2]

```
template<typename T >  
Rcpp::ChildVector< T >::ChildVector (   
    const ChildVector< T > & other ) [inline]
```

Definition at line 35 of file ChildVector.h.

### 6.63.3 Member Function Documentation

#### 6.63.3.1 operator=() [1/3]

```
template<typename T >  
ChildVector& Rcpp::ChildVector< T >::operator= (   
    const ChildVector< T > & other ) [inline]
```

Definition at line 40 of file ChildVector.h.

References Rcpp::ChildVector< T >::i, and Rcpp::ChildVector< T >::parent.



**6.63.3.2 operator=()** [2/3]

```
template<typename T >
ChildVector& Rcpp::ChildVector< T >::operator= (
    const T & other ) [inline]
```

Definition at line 50 of file ChildVector.h.

References Rcpp::ChildVector< T >::i, and Rcpp::ChildVector< T >::parent.

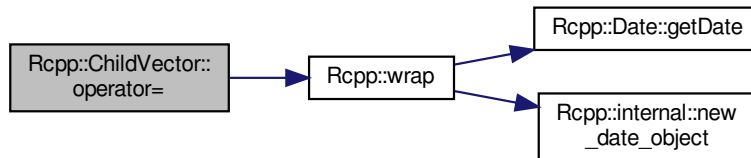
**6.63.3.3 operator=()** [3/3]

```
template<typename T >
template<typename U >
ChildVector& Rcpp::ChildVector< T >::operator= (
    const U & other ) [inline]
```

Definition at line 59 of file ChildVector.h.

References Rcpp::ChildVector< T >::i, Rcpp::ChildVector< T >::parent, and Rcpp::wrap().

Here is the call graph for this function:

**6.63.4 Member Data Documentation****6.63.4.1 i**

```
template<typename T >
R_xlen_t Rcpp::ChildVector< T >::i [private]
```

Definition at line 71 of file ChildVector.h.

Referenced by Rcpp::ChildVector< T >::operator=().

### 6.63.4.2 parent

```
template<typename T >  
SEXPRcpp::ChildVector< T >::parent [private]
```

Definition at line 70 of file ChildVector.h.

Referenced by Rcpp::ChildVector< T >::operator=().

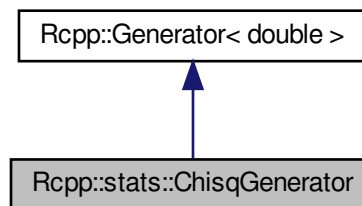
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/ChildVector.h

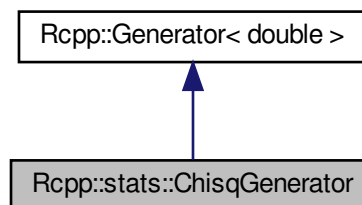
## 6.64 Rcpp::stats::ChisqGenerator Class Reference

```
#include <rchisq.h>
```

Inheritance diagram for Rcpp::stats::ChisqGenerator:



Collaboration diagram for Rcpp::stats::ChisqGenerator:



## Public Member Functions

- [ChisqGenerator](#) (double df\_)
- double [operator\(\)](#) () const

## Private Attributes

- double [df\\_2](#)

## Additional Inherited Members

### 6.64.1 Detailed Description

Definition at line 28 of file rchisq.h.

### 6.64.2 Constructor & Destructor Documentation

#### 6.64.2.1 ChisqGenerator()

```
Rcpp::stats::ChisqGenerator::ChisqGenerator (  
    double df_ ) [inline]
```

Definition at line 31 of file rchisq.h.

### 6.64.3 Member Function Documentation

#### 6.64.3.1 operator>()

```
double Rcpp::stats::ChisqGenerator::operator() ( ) const [inline]
```

Definition at line 33 of file rchisq.h.

References [df\\_2](#).

### 6.64.4 Member Data Documentation

### 6.64.4.1 df\_2

```
double Rcpp::stats::ChisqGenerator::df_2 [private]
```

Definition at line 38 of file rchisq.h.

Referenced by operator()).

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rchisq.h](#)

## 6.65 Rcpp::sugar::clamp\_operator< RTYPE, NA > Struct Template Reference

```
#include <clamp.h>
```

### Public Types

- typedef [Rcpp::traits::storage\\_type< RTYPE >::type STORAGE](#)

### Public Member Functions

- [clamp\\_operator](#) ([STORAGE lhs\\_](#), [STORAGE rhs\\_](#))
- [STORAGE operator\(\)](#) ([STORAGE x](#)) const

### Public Attributes

- [STORAGE lhs](#)
- [STORAGE rhs](#)

### 6.65.1 Detailed Description

```
template<int RTYPE, bool NA>
struct Rcpp::sugar::clamp_operator< RTYPE, NA >
```

Definition at line 29 of file clamp.h.

### 6.65.2 Member Typedef Documentation

### 6.65.2.1 STORAGE

```
template<int RTYPE, bool NA>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::clamp_operator< RTYPE, NA >::STORAGE
```

Definition at line 30 of file clamp.h.

## 6.65.3 Constructor & Destructor Documentation

### 6.65.3.1 clamp\_operator()

```
template<int RTYPE, bool NA>
Rcpp::sugar::clamp_operator< RTYPE, NA >::clamp_operator (
    STORAGE lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 32 of file clamp.h.

## 6.65.4 Member Function Documentation

### 6.65.4.1 operator>()

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::clamp_operator< RTYPE, NA >::operator() (
    STORAGE x ) const [inline]
```

Definition at line 34 of file clamp.h.

References Rcpp::sugar::clamp\_operator< RTYPE, NA >::lhs, and Rcpp::sugar::clamp\_operator< RTYPE, NA >::rhs.

## 6.65.5 Member Data Documentation

### 6.65.5.1 lhs

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::clamp_operator< RTYPE, NA >::lhs
```

Definition at line 37 of file clamp.h.

Referenced by `Rcpp::sugar::clamp_operator< REALSXP, true >::operator()`, and `Rcpp::sugar::clamp_operator< RTYPE, NA >::operator()`.

### 6.65.5.2 rhs

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::clamp_operator< RTYPE, NA >::rhs
```

Definition at line 37 of file clamp.h.

Referenced by `Rcpp::sugar::clamp_operator< REALSXP, true >::operator()`, and `Rcpp::sugar::clamp_operator< RTYPE, NA >::operator()`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/sugar/functions/clamp.h`

## 6.66 Rcpp::sugar::clamp\_operator< REALSXP, true > Struct Reference

```
#include <clamp.h>
```

### Public Member Functions

- `clamp_operator` (double lhs\_, double rhs\_)
- `double operator()` (double x) const

### Public Attributes

- double `lhs`
- double `rhs`

### 6.66.1 Detailed Description

Definition at line 41 of file clamp.h.

## 6.66.2 Constructor & Destructor Documentation

### 6.66.2.1 clamp\_operator()

```
Rcpp::sugar::clamp_operator< REALSXP, true >::clamp_operator (
    double lhs_,
    double rhs_ ) [inline]
```

Definition at line 42 of file clamp.h.

## 6.66.3 Member Function Documentation

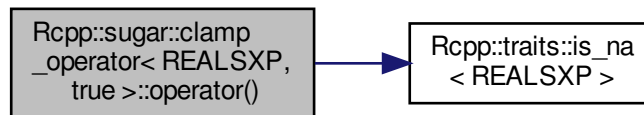
### 6.66.3.1 operator>()

```
double Rcpp::sugar::clamp_operator< REALSXP, true >::operator() (
    double x ) const [inline]
```

Definition at line 44 of file clamp.h.

References [Rcpp::traits::is\\_na< REALSXP >\(\)](#), [Rcpp::sugar::clamp\\_operator< RTYPE, NA >::lhs](#), and [Rcpp::sugar::clamp\\_operator< RTYPE, NA >::rhs](#).

Here is the call graph for this function:



## 6.66.4 Member Data Documentation

### 6.66.4.1 lhs

```
double Rcpp::sugar::clamp_operator< REALSXP, true >::lhs
```

Definition at line 48 of file clamp.h.

### 6.66.4.2 rhs

```
double Rcpp::sugar::clamp_operator< REALSXP, true >::rhs
```

Definition at line 48 of file clamp.h.

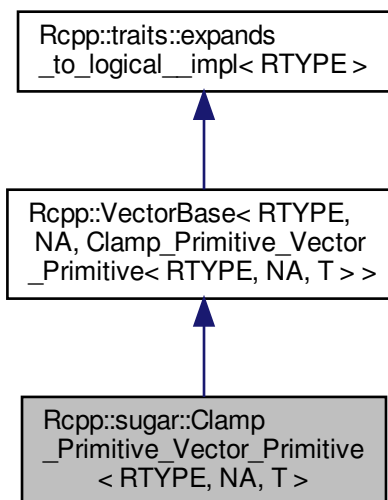
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/clamp.h](#)

## 6.67 Rcpp::sugar::Clamp\_Primitive\_Vector\_Primitive< RTYPE, NA, T > Class Template Reference

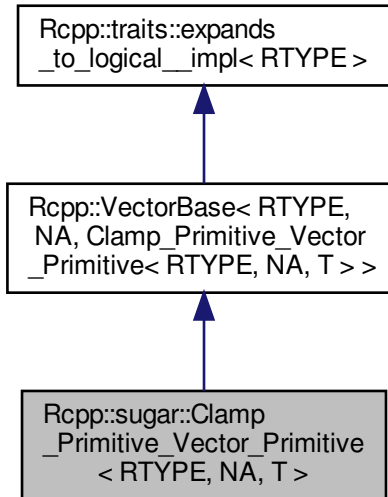
```
#include <clamp.h>
```

Inheritance diagram for Rcpp::sugar::Clamp\_Primitive\_Vector\_Primitive< RTYPE, NA, T >:





Collaboration diagram for Rcpp::sugar::Clamp\_Primitive\_Vector\_Primitive< RTYPE, NA, T >:



## Public Types

- typedef `Rcpp::traits::storage_type < RTYPE >::type` `STORAGE`
- typedef `clamp_operator < RTYPE, NA >` `OPERATOR`

## Public Member Functions

- `Clamp_Primitive_Vector_Primitive (STORAGE lhs_, const T &vec_, STORAGE rhs_)`
- `STORAGE operator[] (R_xlen_t i) const`
- `R_xlen_t size () const`

## Private Attributes

- const T & `vec`
- `OPERATOR op`

### 6.67.1 Detailed Description

```

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >

```

Definition at line 57 of file clamp.h.

## 6.67.2 Member Typedef Documentation

### 6.67.2.1 OPERATOR

```
template<int RTYPE, bool NA, typename T >
typedef clamp_operator<RTYPE,NA> Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >←
::OPERATOR
```

Definition at line 64 of file clamp.h.

### 6.67.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Clamp_Primitive_Vector_Primitive<
RTYPE, NA, T >::STORAGE
```

Definition at line 63 of file clamp.h.

## 6.67.3 Constructor & Destructor Documentation

### 6.67.3.1 Clamp\_Primitive\_Vector\_Primitive()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::Clamp_Primitive_Vector_Primitive (
    STORAGE lhs_,
    const T & vec_,
    STORAGE rhs_ ) [inline]
```

Definition at line 66 of file clamp.h.

## 6.67.4 Member Function Documentation

### 6.67.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >  
STORAGE Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::operator[] (   
    R_xlen_t i ) const [inline]
```

Definition at line 68 of file clamp.h.

References Rcpp::sugar::Clamp\_Primitive\_Vector\_Primitive< RTYPE, NA, T >::op, and Rcpp::sugar::Clamp\_↔  
Primitive\_Vector\_Primitive< RTYPE, NA, T >::vec.

### 6.67.4.2 size()

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 71 of file clamp.h.

References Rcpp::sugar::Clamp\_Primitive\_Vector\_Primitive< RTYPE, NA, T >::vec.

## 6.67.5 Member Data Documentation

### 6.67.5.1 op

```
template<int RTYPE, bool NA, typename T >  
OPERATOR Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::op [private]
```

Definition at line 75 of file clamp.h.

Referenced by Rcpp::sugar::Clamp\_Primitive\_Vector\_Primitive< RTYPE, NA, T >::operator[]().

### 6.67.5.2 vec

```
template<int RTYPE, bool NA, typename T >  
const T& Rcpp::sugar::Clamp_Primitive_Vector_Primitive< RTYPE, NA, T >::vec [private]
```

Definition at line 74 of file clamp.h.

Referenced by Rcpp::sugar::Clamp\_Primitive\_Vector\_Primitive< RTYPE, NA, T >::operator[](), and Rcpp::sugar::↔  
Clamp\_Primitive\_Vector\_Primitive< RTYPE, NA, T >::size().

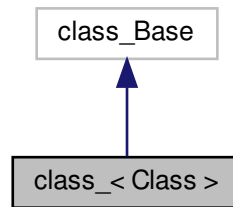
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/clamp.h

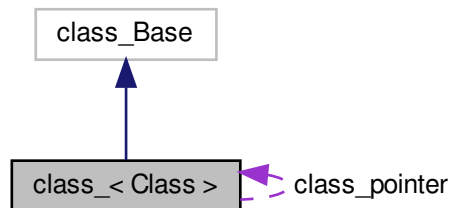
## 6.68 `class_< Class >` Class Template Reference

```
#include <class.h>
```

Inheritance diagram for `class_< Class >`:



Collaboration diagram for `class_< Class >`:



### Public Types

- typedef `class_< Class >` `self`
- typedef `CppMethod< Class >` `method_class`
- typedef `SignedMethod< Class >` `signed_method_class`
- typedef `std::vector< signed_method_class * >` `vec_signed_method`
- typedef `std::map< std::string, vec_signed_method * >` `map_vec_signed_method`
- typedef `std::pair< std::string, vec_signed_method * >` `vec_signed_method_pair`
- typedef `std::map< std::string, method_class * >` `METHOD_MAP`
- typedef `std::pair< const std::string, method_class * >` `PAIR`
- typedef `Rcpp::XPtr< Class >` `XP`
- typedef `CppMethodFinalizer< Class >` `finalizer_class`

- typedef Constructor\_Base< Class > [constructor\\_class](#)
- typedef SignedConstructor< Class > [signed\\_constructor\\_class](#)
- typedef std::vector< [signed\\_constructor\\_class](#) \* > [vec\\_signed\\_constructor](#)
- typedef Factory\_Base< Class > [factory\\_class](#)
- typedef SignedFactory< Class > [signed\\_factory\\_class](#)
- typedef std::vector< [signed\\_factory\\_class](#) \* > [vec\\_signed\\_factory](#)
- typedef CppProperty< Class > [prop\\_class](#)
- typedef std::map< std::string, [prop\\_class](#) \* > [PROPERTY\\_MAP](#)
- typedef std::pair< const std::string, [prop\\_class](#) \* > [PROP\\_PAIR](#)

## Public Member Functions

- [class\\_](#) (const char \*name\_, const char \*doc=0)
- [~class\\_](#) ()
- [self](#) & [AddConstructor](#) ([constructor\\_class](#) \*ctor, ValidConstructor valid, const char \*docstring=0)
- [self](#) & [AddFactory](#) ([factory\\_class](#) \*fact, ValidConstructor valid, const char \*docstring=0)
- [self](#) & [default\\_constructor](#) (const char \*docstring=0, ValidConstructor valid=&yes\_arity < 0 >)
- std::string [get\\_typeinfo\\_name](#) ()
- SEXP [newInstance](#) (SEXP \*args, int nargs)
- bool [has\\_default\\_constructor](#) ()
- SEXP [invoke](#) (SEXP method\_xp, SEXP object, SEXP \*args, int nargs)
- SEXP [invoke\\_void](#) (SEXP method\_xp, SEXP object, SEXP \*args, int nargs)
- SEXP [invoke\\_notvoid](#) (SEXP method\_xp, SEXP object, SEXP \*args, int nargs)
- [self](#) & [AddMethod](#) (const char \*name\_, [method\\_class](#) \*m, ValidMethod valid=&yes, const char \*docstring=0)
- [self](#) & [AddProperty](#) (const char \*name\_, [prop\\_class](#) \*p)
- bool [has\\_method](#) (const std::string &m)
- bool [has\\_property](#) (const std::string &m)
- bool [property\\_is\\_readonly](#) (const std::string &p)
- std::string [property\\_class](#) (const std::string &p)
- Rcpp::CharacterVector [method\\_names](#) ()
- Rcpp::IntegerVector [methods\\_arity](#) ()
- Rcpp::LogicalVector [methods\\_voidness](#) ()
- Rcpp::CharacterVector [property\\_names](#) ()
- Rcpp::List [property\\_classes](#) ()
- Rcpp::CharacterVector [complete](#) ()
- SEXP [getProperty](#) (SEXP field\_xp, SEXP object)
- void [setProperty](#) (SEXP field\_xp, SEXP object, SEXP value)
- Rcpp::List [fields](#) (const [XP\\_Class](#) &class\_xp)
- Rcpp::List [getMethods](#) (const [XP\\_Class](#) &class\_xp, std::string &buffer)
- Rcpp::List [getConstructors](#) (const [XP\\_Class](#) &class\_xp, std::string &buffer)
- [self](#) & [finalizer](#) (void(\*f)(Class \*))
- virtual void [run\\_finalizer](#) (SEXP object)
- void [SetFinalizer](#) ([finalizer\\_class](#) \*f)
- [class\\_](#) ()
- template<typename PARENT >  
[self](#) & [derives](#) (const char \*parent)

## Public Attributes

- [map\\_vec\\_signed\\_method](#) [vec\\_methods](#)
- [PROPERTY\\_MAP](#) [properties](#)
- [finalizer\\_class](#) \* [finalizer\\_pointer](#)
- [int](#) [specials](#)
- [vec\\_signed\\_constructor](#) [constructors](#)
- [vec\\_signed\\_factory](#) [factories](#)
- [self](#) \* [class\\_pointer](#)
- [std::string](#) [typeid\\_name](#)

## Private Member Functions

- [self](#) \* [get\\_instance](#) ()

### 6.68.1 Detailed Description

```
template<typename Class >  
class class_ < Class >
```

Definition at line 26 of file class.h.

### 6.68.2 Member Typedef Documentation

#### 6.68.2.1 constructor\_class

```
template<typename Class >  
typedef Constructor_Base<Class> class_ < Class >::constructor_class
```

Definition at line 42 of file class.h.

#### 6.68.2.2 factory\_class

```
template<typename Class >  
typedef Factory_Base<Class> class_ < Class >::factory_class
```

Definition at line 46 of file class.h.

### 6.68.2.3 finalizer\_class

```
template<typename Class >
typedef CppFinalizer<Class> class_< Class >::finalizer_class
```

Definition at line 40 of file class.h.

### 6.68.2.4 map\_vec\_signed\_method

```
template<typename Class >
typedef std::map<std::string,vec_signed_method*> class_< Class >::map_vec_signed_method
```

Definition at line 33 of file class.h.

### 6.68.2.5 method\_class

```
template<typename Class >
typedef CppMethod<Class> class_< Class >::method_class
```

Definition at line 29 of file class.h.

### 6.68.2.6 METHOD\_MAP

```
template<typename Class >
typedef std::map<std::string,method_class*> class_< Class >::METHOD_MAP
```

Definition at line 36 of file class.h.

### 6.68.2.7 PAIR

```
template<typename Class >
typedef std::pair<const std::string,method_class*> class_< Class >::PAIR
```

Definition at line 37 of file class.h.

### 6.68.2.8 prop\_class

```
template<typename Class >  
typedef CppProperty<Class> class_< Class >::prop_class
```

Definition at line 50 of file class.h.

### 6.68.2.9 PROP\_PAIR

```
template<typename Class >  
typedef std::pair<const std::string,prop_class*> class_< Class >::PROP_PAIR
```

Definition at line 52 of file class.h.

### 6.68.2.10 PROPERTY\_MAP

```
template<typename Class >  
typedef std::map<std::string,prop_class*> class_< Class >::PROPERTY_MAP
```

Definition at line 51 of file class.h.

### 6.68.2.11 self

```
template<typename Class >  
typedef class_<Class> class_< Class >::self
```

Definition at line 28 of file class.h.

### 6.68.2.12 signed\_constructor\_class

```
template<typename Class >  
typedef SignedConstructor<Class> class_< Class >::signed_constructor_class
```

Definition at line 43 of file class.h.



### 6.68.2.13 signed\_factory\_class

```
template<typename Class >  
typedef SignedFactory<Class> class_< Class >::signed_factory_class
```

Definition at line 47 of file class.h.

### 6.68.2.14 signed\_method\_class

```
template<typename Class >  
typedef SignedMethod<Class> class_< Class >::signed_method_class
```

Definition at line 31 of file class.h.

### 6.68.2.15 vec\_signed\_constructor

```
template<typename Class >  
typedef std::vector<signed_constructor_class*> class_< Class >::vec_signed_constructor
```

Definition at line 44 of file class.h.

### 6.68.2.16 vec\_signed\_factory

```
template<typename Class >  
typedef std::vector<signed_factory_class*> class_< Class >::vec_signed_factory
```

Definition at line 48 of file class.h.

### 6.68.2.17 vec\_signed\_method

```
template<typename Class >  
typedef std::vector<signed_method_class*> class_< Class >::vec_signed_method
```

Definition at line 32 of file class.h.

### 6.68.2.18 `vec_signed_method_pair`

```
template<typename Class >  
typedef std::pair<std::string,vec_signed_method*> class_< Class >::vec_signed_method_pair
```

Definition at line 34 of file class.h.

### 6.68.2.19 `XP`

```
template<typename Class >  
typedef Rcpp::XPtr<Class> class_< Class >::XP
```

Definition at line 39 of file class.h.

## 6.68.3 Constructor & Destructor Documentation

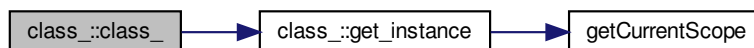
### 6.68.3.1 `class_()` [1/2]

```
template<typename Class >  
class_< Class >::class_ (  
    const char * name_,  
    const char * doc = 0 ) [inline]
```

Definition at line 54 of file class.h.

References `class_< Class >::class_pointer`, and `class_< Class >::get_instance()`.

Here is the call graph for this function:



### 6.68.3.2 ~class\_()

```
template<typename Class >  
class_< Class >::~~class_ ( ) [inline]
```

Definition at line 98 of file class.h.

### 6.68.3.3 class\_() [2/2]

```
template<typename Class >  
class_< Class >::class_ ( ) [inline]
```

Definition at line 481 of file class.h.

## 6.68.4 Member Function Documentation

### 6.68.4.1 AddConstructor()

```
template<typename Class >  
self& class_< Class >::AddConstructor (   
    constructor_class * ctor,  
    ValidConstructor valid,  
    const char * docstring = 0 ) [inline]
```

Definition at line 100 of file class.h.

References class\_< Class >::class\_pointer, and class\_< Class >::constructors.

### 6.68.4.2 AddFactory()

```
template<typename Class >  
self& class_< Class >::AddFactory (   
    factory_class * fact,  
    ValidConstructor valid,  
    const char * docstring = 0 ) [inline]
```

Definition at line 105 of file class.h.

References class\_< Class >::class\_pointer, and class\_< Class >::factories.

### 6.68.4.3 AddMethod()

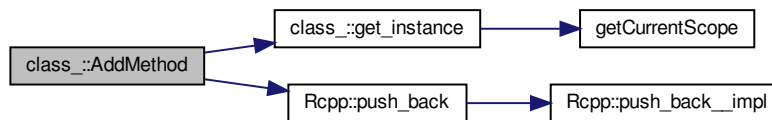
```
template<typename Class >
self& class_< Class >::AddMethod (
    const char * name_,
    method_class * m,
    ValidMethod valid = &yes,
    const char * docstring = 0 ) [inline]
```

Definition at line 239 of file class.h.

References `class_< Class >::get_instance()`, `Rcpp::push_back()`, and `RCPP_DEBUG_MODULE_1`.

Referenced by `class_< Class >::derives()`.

Here is the call graph for this function:



### 6.68.4.4 AddProperty()

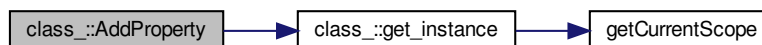
```
template<typename Class >
self& class_< Class >::AddProperty (
    const char * name_,
    prop_class * p ) [inline]
```

Definition at line 251 of file class.h.

References `class_< Class >::get_instance()`, and `class_< Class >::properties`.

Referenced by `class_< Class >::derives()`.

Here is the call graph for this function:



### 6.68.4.5 complete()

```
template<typename Class >
Rcpp::CharacterVector class_< Class >::complete ( ) [inline]
```

Definition at line 368 of file class.h.

References `class_< Class >::properties`, `class_< Class >::specials`, and `class_< Class >::vec_methods`.

### 6.68.4.6 default\_constructor()

```
template<typename Class >
self& class_< Class >::default_constructor (
    const char * docstring = 0,
    ValidConstructor valid = &yes_arity<0> ) [inline]
```

Definition at line 110 of file class.h.

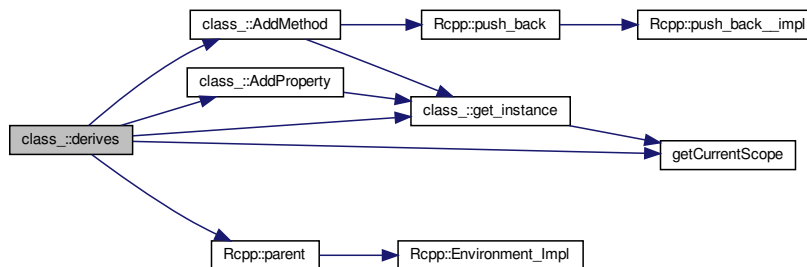
### 6.68.4.7 derives()

```
template<typename Class >
template<typename PARENT >
self& class_< Class >::derives (
    const char * parent ) [inline]
```

Definition at line 487 of file class.h.

References `class_< Class >::AddMethod()`, `class_< Class >::AddProperty()`, `class_< Class >::get_instance()`, `get←CurrentScope()`, and `Rcpp::parent()`.

Here is the call graph for this function:



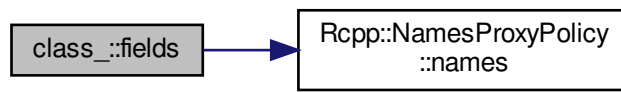
### 6.68.4.8 fields()

```
template<typename Class >
Rcpp::List class_< Class >::fields (
    const XP_Class & class_xp ) [inline]
```

Definition at line 409 of file class.h.

References Rcpp::NamesProxyPolicy< CLASS >::names(), and class\_< Class >::properties.

Here is the call graph for this function:



### 6.68.4.9 finalizer()

```
template<typename Class >
self& class_< Class >::finalizer (
    void(*) (Class *) f ) [inline]
```

Definition at line 455 of file class.h.

References class\_< Class >::SetFinalizer().

Here is the call graph for this function:



#### 6.68.4.10 get\_instance()

```
template<typename Class >
self* class_< Class >::get_instance ( ) [inline], [private]
```

Definition at line 70 of file class.h.

References class\_< Class >::class\_pointer, DEMANGLE, class\_< Class >::finalizer\_pointer, getCurrentScope(), RCPP\_DEBUG\_MODULE\_2, and class\_< Class >::typeinfo\_name.

Referenced by class\_< Class >::AddMethod(), class\_< Class >::AddProperty(), class\_< Class >::class\_(), class\_< Class >::derives(), and class\_< Class >::SetFinalizer().

Here is the call graph for this function:



#### 6.68.4.11 get\_typeinfo\_name()

```
template<typename Class >
std::string class_< Class >::get_typeinfo_name ( ) [inline]
```

Definition at line 119 of file class.h.

References class\_< Class >::typeinfo\_name.

#### 6.68.4.12 getConstructors()

```
template<typename Class >
Rcpp::List class_< Class >::getConstructors (
    const XP_Class & class_xp,
    std::string & buffer ) [inline]
```

Definition at line 441 of file class.h.

References class\_< Class >::constructors.

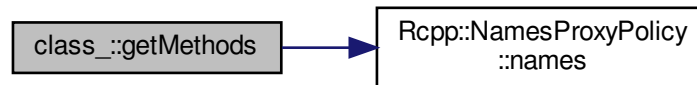
### 6.68.4.13 getMethods()

```
template<typename Class >
Rcpp::List class_< Class >::getMethods (
    const XP_Class & class_xp,
    std::string & buffer ) [inline]
```

Definition at line 422 of file class.h.

References Rcpp::NamesProxyPolicy< CLASS >::names(), RCPP\_DEBUG\_MODULE, and class\_< Class >::vec\_↔ methods.

Here is the call graph for this function:



### 6.68.4.14 getProperty()

```
template<typename Class >
SEXP class_< Class >::getProperty (
    SEXP field_xp,
    SEXP object ) [inline]
```

Definition at line 394 of file class.h.

References BEGIN\_RCPP, and END\_RCPP.

### 6.68.4.15 has\_default\_constructor()

```
template<typename Class >
bool class_< Class >::has_default_constructor ( ) [inline]
```

Definition at line 151 of file class.h.

References class\_< Class >::constructors, and class\_< Class >::factories.



#### 6.68.4.16 `has_method()`

```
template<typename Class >
bool class_< Class >::has_method (
    const std::string & m ) [inline]
```

Definition at line 259 of file `class.h`.

References `class_< Class >::vec_methods`.

#### 6.68.4.17 `has_property()`

```
template<typename Class >
bool class_< Class >::has_property (
    const std::string & m ) [inline]
```

Definition at line 262 of file `class.h`.

References `class_< Class >::properties`.

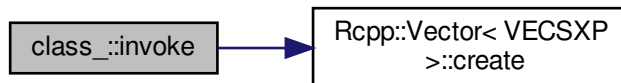
#### 6.68.4.18 `invoke()`

```
template<typename Class >
SEXP class_< Class >::invoke (
    SEXP method_xp,
    SEXP object,
    SEXP * args,
    int nargs ) [inline]
```

Definition at line 167 of file `class.h`.

References `BEGIN_RCPP`, `Rcpp::Vector< VECSXP >::create()`, and `END_RCPP`.

Here is the call graph for this function:



#### 6.68.4.19 `invoke_notvoid()`

```
template<typename Class >
SEXP class_< Class >::invoke_notvoid (
    SEXP method_xp,
    SEXP object,
    SEXP * args,
    int nargs ) [inline]
```

Definition at line 216 of file `class.h`.

References `BEGIN_RCPP`, and `END_RCPP`.

#### 6.68.4.20 `invoke_void()`

```
template<typename Class >
SEXP class_< Class >::invoke_void (
    SEXP method_xp,
    SEXP object,
    SEXP * args,
    int nargs ) [inline]
```

Definition at line 194 of file `class.h`.

References `BEGIN_RCPP`, and `END_RCPP`.

#### 6.68.4.21 `method_names()`

```
template<typename Class >
Rcpp::CharacterVector class_< Class >::method_names ( ) [inline]
```

Definition at line 276 of file `class.h`.

References `class_< Class >::vec_methods`.

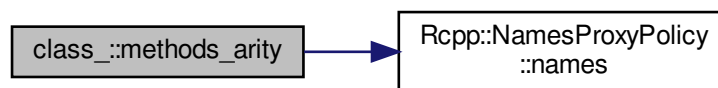
#### 6.68.4.22 methods\_arity()

```
template<typename Class >  
Rcpp::IntegerVector class_< Class >::methods_arity ( ) [inline]
```

Definition at line 296 of file class.h.

References Rcpp::NamesProxyPolicy< CLASS >::names(), and class\_< Class >::vec\_methods.

Here is the call graph for this function:



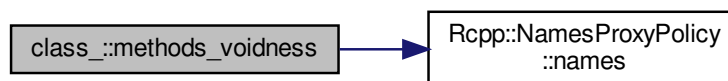
#### 6.68.4.23 methods\_voidness()

```
template<typename Class >  
Rcpp::LogicalVector class_< Class >::methods_voidness ( ) [inline]
```

Definition at line 320 of file class.h.

References Rcpp::NamesProxyPolicy< CLASS >::names(), and class\_< Class >::vec\_methods.

Here is the call graph for this function:



**6.68.4.24 newInstance()**

```
template<typename Class >
SEXP class_< Class >::newInstance (
    SEXP * args,
    int nargs ) [inline]
```

Definition at line 123 of file class.h.

References BEGIN\_RCPP, class\_< Class >::constructors, END\_RCPP, and class\_< Class >::factories.

**6.68.4.25 property\_class()**

```
template<typename Class >
std::string class_< Class >::property_class (
    const std::string & p ) [inline]
```

Definition at line 270 of file class.h.

References class\_< Class >::properties.

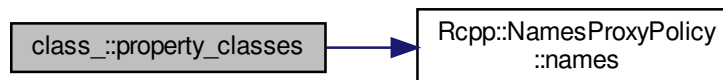
**6.68.4.26 property\_classes()**

```
template<typename Class >
Rcpp::List class_< Class >::property_classes ( ) [inline]
```

Definition at line 355 of file class.h.

References Rcpp::NamesProxyPolicy< CLASS >::names(), and class\_< Class >::properties.

Here is the call graph for this function:



#### 6.68.4.27 property\_is\_readonly()

```
template<typename Class >
bool class_< Class >::property_is_readonly (
    const std::string & p ) [inline]
```

Definition at line 265 of file class.h.

References class\_< Class >::properties.

#### 6.68.4.28 property\_names()

```
template<typename Class >
Rcpp::CharacterVector class_< Class >::property_names ( ) [inline]
```

Definition at line 345 of file class.h.

References class\_< Class >::properties.

#### 6.68.4.29 run\_finalizer()

```
template<typename Class >
virtual void class_< Class >::run_finalizer (
    SEXP object ) [inline], [virtual]
```

Definition at line 460 of file class.h.

References class\_< Class >::finalizer\_pointer.

#### 6.68.4.30 SetFinalizer()

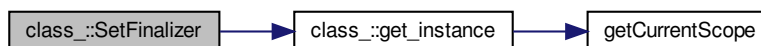
```
template<typename Class >
void class_< Class >::SetFinalizer (
    finalizer_class * f ) [inline]
```

Definition at line 464 of file class.h.

References class\_< Class >::get\_instance().

Referenced by class\_< Class >::finalizer().

Here is the call graph for this function:



### 6.68.4.31 setProperty()

```
template<typename Class >
void class_< Class >::setProperty (
    SEXP field_xp,
    SEXP object,
    SEXP value ) [inline]
```

Definition at line 401 of file class.h.

References BEGIN\_RCPP, and VOID\_END\_RCPP.

## 6.68.5 Member Data Documentation

### 6.68.5.1 class\_pointer

```
template<typename Class >
self* class_< Class >::class_pointer
```

Definition at line 477 of file class.h.

Referenced by class\_< Class >::AddConstructor(), class\_< Class >::AddFactory(), class\_< Class >::class\_(), and class\_< Class >::get\_instance().

### 6.68.5.2 constructors

```
template<typename Class >
vec_signed_constructor class_< Class >::constructors
```

Definition at line 475 of file class.h.

Referenced by class\_< Class >::AddConstructor(), class\_< Class >::getConstructors(), class\_< Class >::has\_↔ default\_constructor(), and class\_< Class >::newInstance().

### 6.68.5.3 factories

```
template<typename Class >
vec_signed_factory class_< Class >::factories
```

Definition at line 476 of file class.h.

Referenced by class\_< Class >::AddFactory(), class\_< Class >::has\_default\_constructor(), and class\_< Class >↔ ::newInstance().

#### 6.68.5.4 finalizer\_pointer

```
template<typename Class >  
finalizer_class* class_< Class >::finalizer_pointer
```

Definition at line 473 of file class.h.

Referenced by class\_< Class >::get\_instance(), and class\_< Class >::run\_finalizer().

#### 6.68.5.5 properties

```
template<typename Class >  
PROPERTY_MAP class_< Class >::properties
```

Definition at line 471 of file class.h.

Referenced by class\_< Class >::AddProperty(), class\_< Class >::complete(), class\_< Class >::fields(), class\_< Class >::has\_property(), class\_< Class >::property\_class(), class\_< Class >::property\_classes(), class\_< Class >::property\_is\_readonly(), and class\_< Class >::property\_names().

#### 6.68.5.6 specials

```
template<typename Class >  
int class_< Class >::specials
```

Definition at line 474 of file class.h.

Referenced by class\_< Class >::complete().

#### 6.68.5.7 typeid\_name

```
template<typename Class >  
std::string class_< Class >::typeid_name
```

Definition at line 478 of file class.h.

Referenced by class\_< Class >::get\_instance(), and class\_< Class >::get\_typeinfo\_name().

### 6.68.5.8 `vec_methods`

```
template<typename Class >
map_vec_signed_method class_< Class >::vec_methods
```

Definition at line 470 of file class.h.

Referenced by `class_< Class >::complete()`, `class_< Class >::getMethods()`, `class_< Class >::has_method()`, `class_< Class >::method_names()`, `class_< Class >::methods_arity()`, and `class_< Class >::methods_voidness()`.

The documentation for this class was generated from the following file:

- `inst/include/Rcpp/module/class.h`

## 6.69 `Rcpp::class_Base` Class Reference

```
#include <class_Base.h>
```

### Public Types

- typedef `Rcpp::XPtr< class_Base > XP_Class`
- typedef `std::map< std::string, int > ENUM`
- typedef `std::map< std::string, ENUM > ENUM_MAP`
- typedef `ENUM_MAP::value_type ENUM_MAP_PAIR`

### Public Member Functions

- `class_Base ()`
- `class_Base (const char *name_, const char *doc)`
- virtual `Rcpp::List fields (const XP_Class &)`
- virtual `Rcpp::List getMethods (const XP_Class &, std::string &)`
- virtual `Rcpp::List getConstructors (const XP_Class &, std::string &)`
- virtual void `run_finalizer (SEXP)`
- virtual bool `has_default_constructor ()`
- virtual bool `has_method (const std::string &)`
- virtual bool `has_property (const std::string &)`
- virtual SEXP `newInstance (SEXP *, int)`
- virtual SEXP `invoke (SEXP, SEXP, SEXP *, int)`
- virtual SEXP `invoke_void (SEXP, SEXP, SEXP *, int)`
- virtual SEXP `invoke_notvoid (SEXP, SEXP, SEXP *, int)`
- virtual `Rcpp::CharacterVector method_names ()`
- virtual `Rcpp::CharacterVector property_names ()`
- virtual bool `property_is_readonly (const std::string &)`
- virtual `std::string property_class (const std::string &)`
- virtual `Rcpp::IntegerVector methods_arity ()`
- virtual `Rcpp::LogicalVector methods_voidness ()`
- virtual `Rcpp::List property_classes ()`
- virtual `Rcpp::CharacterVector complete ()`
- virtual `~class_Base ()`
- virtual SEXP `getProperty (SEXP, SEXP)`
- virtual void `setProperty (SEXP, SEXP, SEXP)`
- virtual `std::string get_typeinfo_name ()`
- bool `has_typeinfo_name (const std::string &name_)`
- void `add_enum (const std::string &enum_name, const std::map< std::string, int > &value)`



## Public Attributes

- `std::string` [name](#)
- `std::string` [docstring](#)
- [ENUM\\_MAP](#) `enums`
- `std::vector< std::string >` [parents](#)

### 6.69.1 Detailed Description

Definition at line 26 of file `class_Base.h`.

### 6.69.2 Member Typedef Documentation

#### 6.69.2.1 ENUM

```
typedef std::map< std::string, int > Rcpp::class_Base::ENUM
```

Definition at line 88 of file `class_Base.h`.

#### 6.69.2.2 ENUM\_MAP

```
typedef std::map< std::string, ENUM > Rcpp::class_Base::ENUM_MAP
```

Definition at line 89 of file `class_Base.h`.

#### 6.69.2.3 ENUM\_MAP\_PAIR

```
typedef ENUM_MAP::value_type Rcpp::class_Base::ENUM_MAP_PAIR
```

Definition at line 90 of file `class_Base.h`.

#### 6.69.2.4 XP\_Class

```
typedef Rcpp::XPtr<class_Base> Rcpp::class_Base::XP_Class
```

Definition at line 28 of file `class_Base.h`.

## 6.69.3 Constructor & Destructor Documentation

### 6.69.3.1 `class_Base()` [1/2]

```
Rcpp::class_Base::class_Base ( ) [inline]
```

Definition at line 30 of file `class_Base.h`.

### 6.69.3.2 `class_Base()` [2/2]

```
Rcpp::class_Base::class_Base (
    const char * name_,
    const char * doc ) [inline]
```

Definition at line 31 of file `class_Base.h`.

### 6.69.3.3 `~class_Base()`

```
virtual Rcpp::class_Base::~~class_Base ( ) [inline], [virtual]
```

Definition at line 69 of file `class_Base.h`.

## 6.69.4 Member Function Documentation

### 6.69.4.1 `add_enum()`

```
void Rcpp::class_Base::add_enum (
    const std::string & enum_name,
    const std::map< std::string, int > & value ) [inline]
```

Definition at line 81 of file `class_Base.h`.

References `enums`.

Referenced by `Rcpp::Module::add_enum()`.

#### 6.69.4.2 complete()

```
virtual Rcpp::CharacterVector Rcpp::class_Base::complete ( ) [inline], [virtual]
```

Definition at line 68 of file class\_Base.h.

#### 6.69.4.3 fields()

```
virtual Rcpp::List Rcpp::class_Base::fields (
    const XP_Class & ) [inline], [virtual]
```

Definition at line 34 of file class\_Base.h.

Referenced by Rcpp::CppClass::CppClass().

#### 6.69.4.4 get\_typeinfo\_name()

```
virtual std::string Rcpp::class_Base::get_typeinfo_name ( ) [inline], [virtual]
```

Definition at line 77 of file class\_Base.h.

Referenced by Rcpp::CppClass::CppClass(), and has\_typeinfo\_name().

#### 6.69.4.5 getConstructors()

```
virtual Rcpp::List Rcpp::class_Base::getConstructors (
    const XP_Class & ,
    std::string & ) [inline], [virtual]
```

Definition at line 36 of file class\_Base.h.

Referenced by Rcpp::CppClass::CppClass().

#### 6.69.4.6 getMethods()

```
virtual Rcpp::List Rcpp::class_Base::getMethods (
    const XP_Class & ,
    std::string & ) [inline], [virtual]
```

Definition at line 35 of file class\_Base.h.

Referenced by Rcpp::CppClass::CppClass().

#### 6.69.4.7 getProperty()

```
virtual SEXP Rcpp::class_Base::getProperty (
    SEXP ,
    SEXP ) [inline], [virtual]
```

Definition at line 71 of file class\_Base.h.

#### 6.69.4.8 has\_default\_constructor()

```
virtual bool Rcpp::class_Base::has_default_constructor ( ) [inline], [virtual]
```

Definition at line 40 of file class\_Base.h.

#### 6.69.4.9 has\_method()

```
virtual bool Rcpp::class_Base::has_method (
    const std::string & ) [inline], [virtual]
```

Definition at line 41 of file class\_Base.h.

#### 6.69.4.10 has\_property()

```
virtual bool Rcpp::class_Base::has_property (
    const std::string & ) [inline], [virtual]
```

Definition at line 44 of file class\_Base.h.

#### 6.69.4.11 has\_typeinfo\_name()

```
bool Rcpp::class_Base::has_typeinfo_name (
    const std::string & name_ ) [inline]
```

Definition at line 78 of file class\_Base.h.

References `get_typeinfo_name()`.

Here is the call graph for this function:



#### 6.69.4.12 invoke()

```
virtual SEXP Rcpp::class_Base::invoke (
    SEXP ,
    SEXP ,
    SEXP * ,
    int ) [inline], [virtual]
```

Definition at line 50 of file class\_Base.h.

#### 6.69.4.13 invoke\_notvoid()

```
virtual SEXP Rcpp::class_Base::invoke_notvoid (
    SEXP ,
    SEXP ,
    SEXP * ,
    int ) [inline], [virtual]
```

Definition at line 56 of file class\_Base.h.

#### 6.69.4.14 invoke\_void()

```
virtual SEXP Rcpp::class_Base::invoke_void (
    SEXP ,
    SEXP ,
    SEXP * ,
    int ) [inline], [virtual]
```

Definition at line 53 of file class\_Base.h.

#### 6.69.4.15 method\_names()

```
virtual Rcpp::CharacterVector Rcpp::class_Base::method_names ( ) [inline], [virtual]
```

Definition at line 60 of file class\_Base.h.

#### 6.69.4.16 methods\_arity()

```
virtual Rcpp::IntegerVector Rcpp::class_Base::methods_arity ( ) [inline], [virtual]
```

Definition at line 64 of file class\_Base.h.

#### 6.69.4.17 methods\_voidness()

```
virtual Rcpp::LogicalVector Rcpp::class_Base::methods_voidness ( ) [inline], [virtual]
```

Definition at line 65 of file class\_Base.h.

#### 6.69.4.18 newInstance()

```
virtual SEXP Rcpp::class_Base::newInstance (
    SEXP * ,
    int ) [inline], [virtual]
```

Definition at line 47 of file class\_Base.h.

#### 6.69.4.19 property\_class()

```
virtual std::string Rcpp::class_Base::property_class (
    const std::string & ) [inline], [virtual]
```

Definition at line 63 of file class\_Base.h.

#### 6.69.4.20 property\_classes()

```
virtual Rcpp::List Rcpp::class_Base::property_classes ( ) [inline], [virtual]
```

Definition at line 66 of file class\_Base.h.

#### 6.69.4.21 property\_is\_readonly()

```
virtual bool Rcpp::class_Base::property_is_readonly (
    const std::string & ) [inline], [virtual]
```

Definition at line 62 of file class\_Base.h.

#### 6.69.4.22 property\_names()

```
virtual Rcpp::CharacterVector Rcpp::class_Base::property_names ( ) [inline], [virtual]
```

Definition at line 61 of file class\_Base.h.

#### 6.69.4.23 run\_finalizer()

```
virtual void Rcpp::class_Base::run_finalizer (
    SEXP ) [inline], [virtual]
```

Definition at line 38 of file class\_Base.h.

#### 6.69.4.24 setProperty()

```
virtual void Rcpp::class_Base::setProperty (
    SEXP ,
    SEXP ,
    SEXP ) [inline], [virtual]
```

Definition at line 74 of file class\_Base.h.

### 6.69.5 Member Data Documentation

#### 6.69.5.1 docstring

```
std::string Rcpp::class_Base::docstring
```

Definition at line 86 of file class\_Base.h.

Referenced by Rcpp::CppClass::CppClass().

### 6.69.5.2 enums

`ENUM_MAP Rcpp::class_Base::enums`

Definition at line 91 of file `class_Base.h`.

Referenced by `add_enum()`, and `Rcpp::CppClass::CppClass()`.

### 6.69.5.3 name

`std::string Rcpp::class_Base::name`

Definition at line 85 of file `class_Base.h`.

Referenced by `Rcpp::CppClass::CppClass()`.

### 6.69.5.4 parents

`std::vector<std::string> Rcpp::class_Base::parents`

Definition at line 92 of file `class_Base.h`.

Referenced by `Rcpp::CppClass::CppClass()`.

The documentation for this class was generated from the following file:

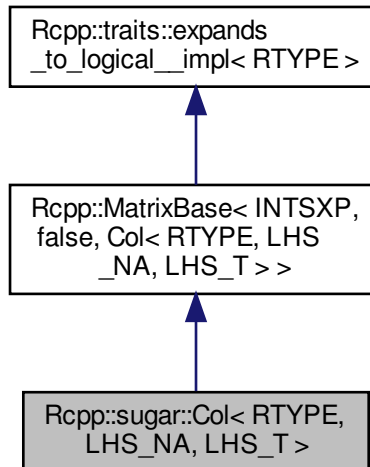
- [inst/include/Rcpp/module/class\\_Base.h](#)



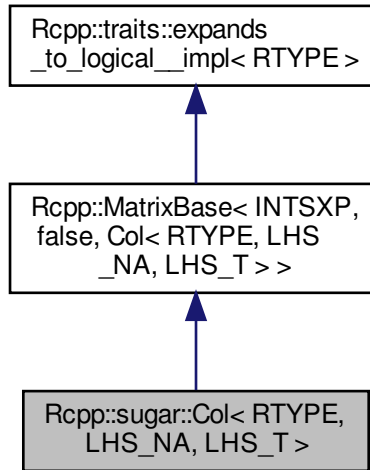
## 6.70 Rcpp::sugar::Col< RTYPE, LHS\_NA, LHS\_T > Class Template Reference

```
#include <col.h>
```

Inheritance diagram for Rcpp::sugar::Col< RTYPE, LHS\_NA, LHS\_T >:



Collaboration diagram for `Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >`:



## Public Types

- typedef `Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T >` `LHS_TYPE`

## Public Member Functions

- `Col` (const `LHS_TYPE` &lhs)
- `operator()` (int i, int j) const
- `R_xlen_t size` () const
- `int nrow` () const
- `int ncol` () const

## Private Attributes

- `int nr`
- `int nc`

## Additional Inherited Members

### 6.70.1 Detailed Description

```

template<int RTYPE, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >
  
```

Definition at line 29 of file col.h.

## 6.70.2 Member Typedef Documentation

### 6.70.2.1 LHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef Rcpp::MatrixBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::LHS_TYPE
```

Definition at line 35 of file col.h.

## 6.70.3 Constructor & Destructor Documentation

### 6.70.3.1 Col()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::Col (
    const LHS_TYPE & lhs ) [inline]
```

Definition at line 37 of file col.h.

## 6.70.4 Member Function Documentation

### 6.70.4.1 ncol()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::ncol ( ) const [inline]
```

Definition at line 45 of file col.h.

References Rcpp::sugar::Col< RTYPE, LHS\_NA, LHS\_T >::nc.

#### 6.70.4.2 nrow()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::nrow ( ) const [inline]
```

Definition at line 44 of file col.h.

References Rcpp::sugar::Col< RTYPE, LHS\_NA, LHS\_T >::nr.

#### 6.70.4.3 operator>()()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::operator() (
    int i,
    int j ) const [inline]
```

Definition at line 39 of file col.h.

#### 6.70.4.4 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::size ( ) const [inline]
```

Definition at line 43 of file col.h.

References Rcpp::sugar::Col< RTYPE, LHS\_NA, LHS\_T >::nc, and Rcpp::sugar::Col< RTYPE, LHS\_NA, LHS\_T >↵  
::nr.

### 6.70.5 Member Data Documentation

#### 6.70.5.1 nc

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::nc [private]
```

Definition at line 48 of file col.h.

Referenced by Rcpp::sugar::Col< RTYPE, LHS\_NA, LHS\_T >::ncol(), and Rcpp::sugar::Col< RTYPE, LHS\_NA, LHS\_T >::size().

## 6.70.5.2 nr

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Col< RTYPE, LHS_NA, LHS_T >::nr [private]
```

Definition at line 48 of file col.h.

Referenced by Rcpp::sugar::Col< RTYPE, LHS\_NA, LHS\_T >::nrow(), and Rcpp::sugar::Col< RTYPE, LHS\_NA, LHS\_T >::size().

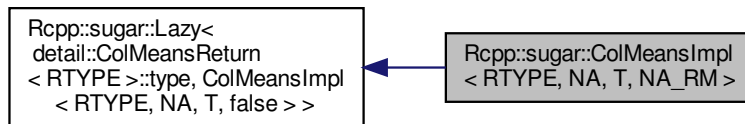
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/col.h

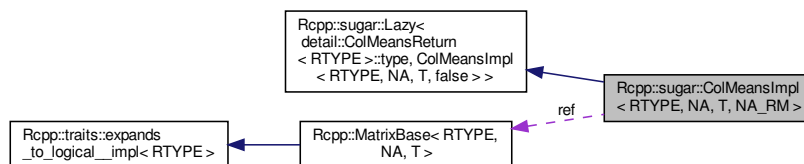
## 6.71 Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA\_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA\_RM >:



Collaboration diagram for Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA\_RM >:



### Public Member Functions

- `ColMeansImpl` (const `MatrixBase< RTYPE, NA, T >` &ref\_)
- `return_vector get` () const

## Private Types

- typedef [detail::ColMeansReturn](#)< RTYPE > [return\\_traits](#)
- typedef [return\\_traits::type](#) [return\\_vector](#)
- typedef [traits::storage\\_type](#)< [return\\_traits::rtype](#) >::type [stored\\_type](#)

## Private Attributes

- const [MatrixBase](#)< RTYPE, NA, T > & [ref](#)

### 6.71.1 Detailed Description

```
template<int RTYPE, bool NA, typename T, bool NA_RM = false>
class Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >
```

Definition at line 716 of file rowSums.h.

### 6.71.2 Member Typedef Documentation

#### 6.71.2.1 return\_traits

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef detail::ColMeansReturn<RTYPE> Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::return\_traits
[private]
```

Definition at line 721 of file rowSums.h.

#### 6.71.2.2 return\_vector

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef return\_traits::type Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::return\_vector [private]
```

Definition at line 722 of file rowSums.h.

### 6.71.2.3 stored\_type

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::ColMeansImpl< RTYPE, NA, T,
NA_RM >::stored_type [private]
```

Definition at line 723 of file rowSums.h.

## 6.71.3 Constructor & Destructor Documentation

### 6.71.3.1 ColMeansImpl()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::ColMeansImpl (
    const MatrixBase< RTYPE, NA, T > & ref_ ) [inline]
```

Definition at line 726 of file rowSums.h.

## 6.71.4 Member Function Documentation

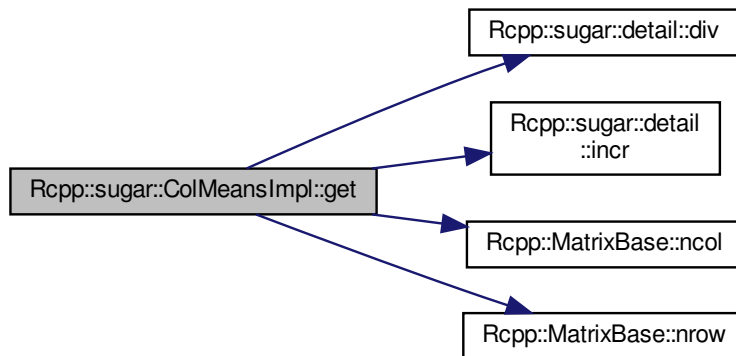
### 6.71.4.1 get()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
return_vector Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::get ( ) const [inline]
```

Definition at line 730 of file rowSums.h.

References Rcpp::sugar::detail::div(), Rcpp::sugar::detail::incr(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow(), and Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA\_RM >::ref.

Here is the call graph for this function:



## 6.71.5 Member Data Documentation

### 6.71.5.1 ref

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::ref [private]
```

Definition at line 719 of file rowSums.h.

Referenced by `Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::get()`, and `Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get()`.

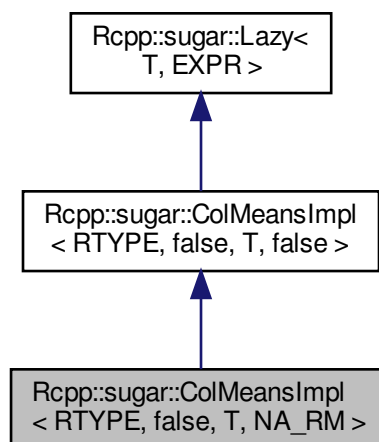
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/rowSums.h`

## 6.72 Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA\_RM > Class Template Reference

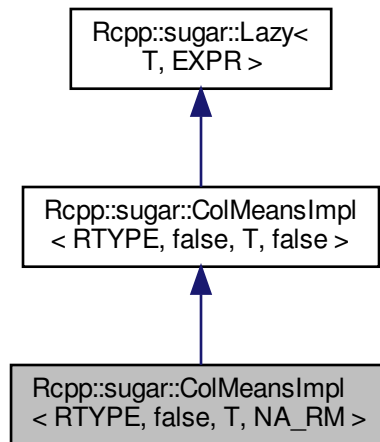
```
#include <rowSums.h>
```

Inheritance diagram for `Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA_RM >`:





Collaboration diagram for Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA\_RM >:



## Additional Inherited Members

### 6.72.1 Detailed Description

```
template<int RTYPE, typename T, bool NA_RM>
class Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA_RM >
```

Definition at line 915 of file rowSums.h.

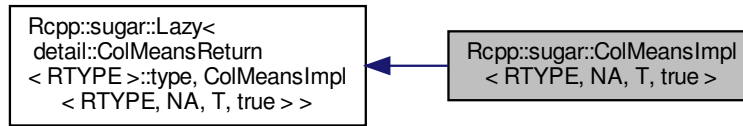
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[rowSums.h](#)

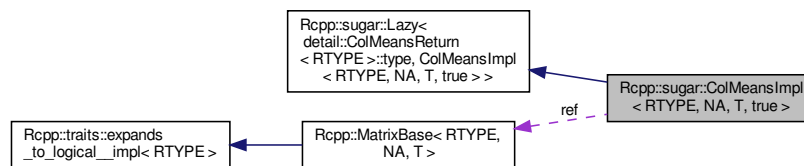
## 6.73 Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for `Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >`:



Collaboration diagram for `Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >`:



## Public Member Functions

- `ColMeansImpl` (const `MatrixBase< RTYPE, NA, T >` &ref\_)
- `return_vector` `get` () const

## Private Types

- typedef `detail::ColMeansReturn< RTYPE >` `return_traits`
- typedef `return_traits::type` `return_vector`
- typedef `traits::storage_type< return_traits::rtype >::type` `stored_type`

## Private Attributes

- const `MatrixBase< RTYPE, NA, T >` & `ref`

### 6.73.1 Detailed Description

```

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >
  
```

Definition at line 812 of file `rowSums.h`.

## 6.73.2 Member Typedef Documentation

### 6.73.2.1 return\_traits

```
template<int RTYPE, bool NA, typename T >
typedef detail::ColMeansReturn<RTYPE> Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::return_traits
[private]
```

Definition at line 817 of file rowSums.h.

### 6.73.2.2 return\_vector

```
template<int RTYPE, bool NA, typename T >
typedef return_traits::type Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::return_vector [private]
```

Definition at line 818 of file rowSums.h.

### 6.73.2.3 stored\_type

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::ColMeansImpl< RTYPE, NA, T,
true >::stored_type [private]
```

Definition at line 819 of file rowSums.h.

## 6.73.3 Constructor & Destructor Documentation

### 6.73.3.1 ColMeansImpl()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::ColMeansImpl (
    const MatrixBase< RTYPE, NA, T > & ref_ ) [inline]
```

Definition at line 822 of file rowSums.h.

## 6.73.4 Member Function Documentation

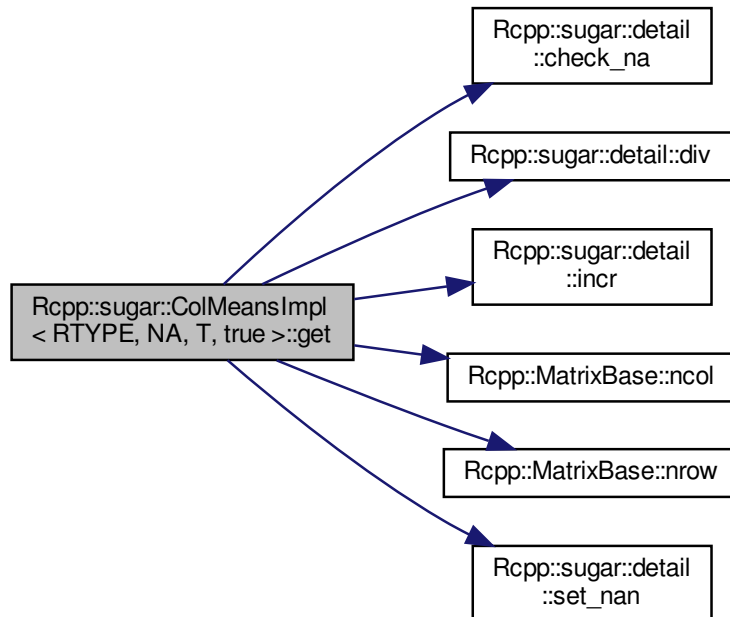
### 6.73.4.1 get()

```
template<int RTYPE, bool NA, typename T >
return_vector Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get ( ) const [inline]
```

Definition at line 826 of file rowSums.h.

References `Rcpp::sugar::detail::check_na()`, `Rcpp::sugar::detail::div()`, `Rcpp::sugar::detail::incr()`, `Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol()`, `Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow()`, `Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA_RM >::ref`, and `Rcpp::sugar::detail::set_nan()`.

Here is the call graph for this function:



## 6.73.5 Member Data Documentation

## 6.73.5.1 ref

```
template<int RTYPE, bool NA, typename T >
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::ref [private]
```

Definition at line 815 of file rowSums.h.

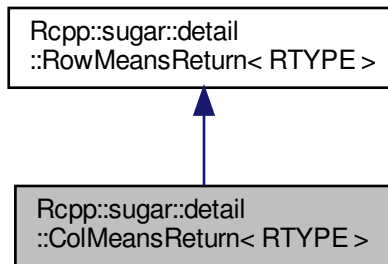
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

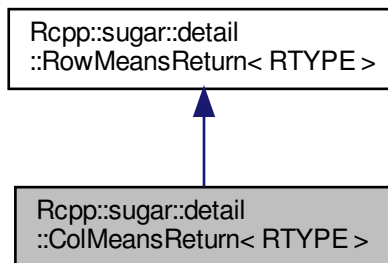
## 6.74 Rcpp::sugar::detail::ColMeansReturn< RTYPE > Struct Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::detail::ColMeansReturn< RTYPE >:



Collaboration diagram for Rcpp::sugar::detail::ColMeansReturn< RTYPE >:



## Additional Inherited Members

### 6.74.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::sugar::detail::ColMeansReturn< RTYPE >
```

Definition at line 115 of file rowSums.h.

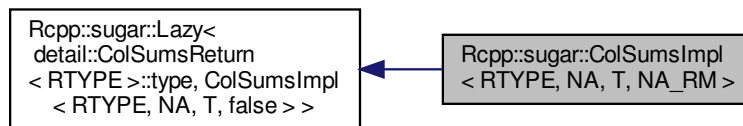
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/[rowSums.h](#)

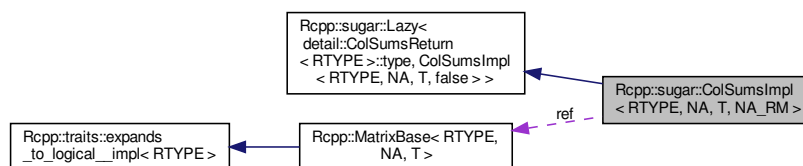
## 6.75 Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA\_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA\_RM >:



Collaboration diagram for Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA\_RM >:



## Public Member Functions

- [ColSumsImpl](#) (const [MatrixBase](#)< RTYPE, NA, T > &ref\_)
- [return\\_vector get](#) () const

## Private Types

- typedef [detail::ColSumsReturn< RTYPE > return\\_traits](#)
- typedef [return\\_traits::type return\\_vector](#)
- typedef [traits::storage\\_type< return\\_traits::rtype >::type stored\\_type](#)

## Private Attributes

- const [MatrixBase< RTYPE, NA, T > & ref](#)

### 6.75.1 Detailed Description

```
template<int RTYPE, bool NA, typename T, bool NA_RM = false>
class Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >
```

Definition at line 320 of file rowSums.h.

### 6.75.2 Member Typedef Documentation

#### 6.75.2.1 return\_traits

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef detail::ColSumsReturn<RTYPE> Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA\_RM >::return\_traits
[private]
```

Definition at line 325 of file rowSums.h.

#### 6.75.2.2 return\_vector

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef return\_traits::type Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA\_RM >::return\_vector [private]
```

Definition at line 326 of file rowSums.h.

### 6.75.2.3 stored\_type

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::ColSumsImpl< RTYPE, NA, T,
NA_RM >::stored_type [private]
```

Definition at line 327 of file rowSums.h.

## 6.75.3 Constructor & Destructor Documentation

### 6.75.3.1 ColSumsImpl()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::ColSumsImpl (
    const MatrixBase< RTYPE, NA, T > & ref_ ) [inline]
```

Definition at line 330 of file rowSums.h.

## 6.75.4 Member Function Documentation

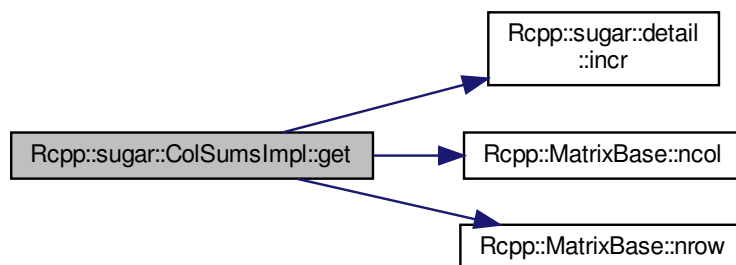
### 6.75.4.1 get()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
return_vector Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::get ( ) const [inline]
```

Definition at line 334 of file rowSums.h.

References `Rcpp::sugar::detail::incr()`, `Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol()`, `Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow()`, and `Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::ref`.

Here is the call graph for this function:





## 6.75.5 Member Data Documentation

### 6.75.5.1 ref

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::ref [private]
```

Definition at line 323 of file rowSums.h.

Referenced by Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA\_RM >::get(), and Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get().

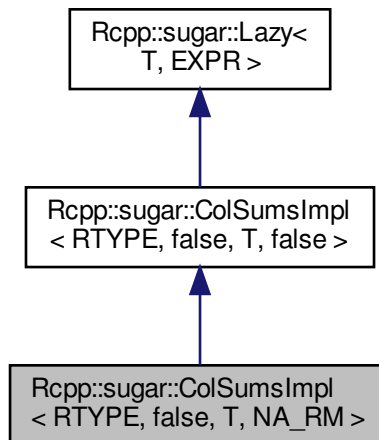
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

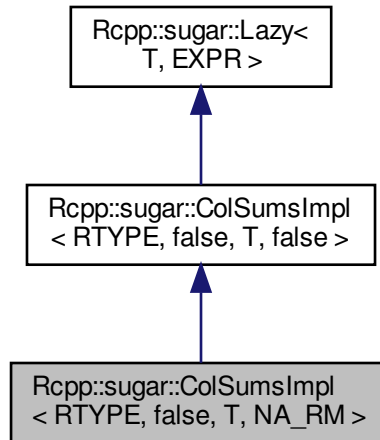
## 6.76 Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA\_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA\_RM >:



Collaboration diagram for `Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA_RM >`:



## Additional Inherited Members

### 6.76.1 Detailed Description

```

template<int RTYPE, typename T, bool NA_RM>
class Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA_RM >

```

Definition at line 494 of file `rowSums.h`.

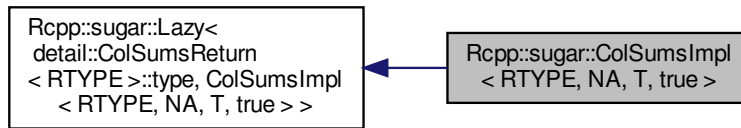
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/rowSums.h](#)

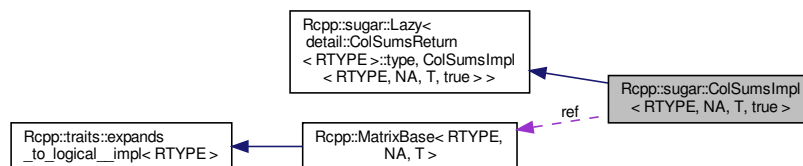
## 6.77 `Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >` Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >:



Collaboration diagram for Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >:



## Public Member Functions

- `ColSumsImpl` (const `MatrixBase< RTYPE, NA, T >` &ref\_)
- `return_vector` `get` () const

## Private Types

- typedef `detail::ColSumsReturn< RTYPE >` `return_traits`
- typedef `return_traits::type` `return_vector`
- typedef `traits::storage_type< return_traits::rtype >` `stored_type`

## Private Attributes

- const `MatrixBase< RTYPE, NA, T >` & `ref`

### 6.77.1 Detailed Description

```

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >
  
```

Definition at line 410 of file `rowSums.h`.

## 6.77.2 Member Typedef Documentation

### 6.77.2.1 return\_traits

```
template<int RTYPE, bool NA, typename T >
typedef detail::ColSumsReturn<RTYPE> Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::return_traits
[private]
```

Definition at line 415 of file rowSums.h.

### 6.77.2.2 return\_vector

```
template<int RTYPE, bool NA, typename T >
typedef return_traits::type Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::return_vector [private]
```

Definition at line 416 of file rowSums.h.

### 6.77.2.3 stored\_type

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::ColSumsImpl< RTYPE, NA, T,
true >::stored_type [private]
```

Definition at line 417 of file rowSums.h.

## 6.77.3 Constructor & Destructor Documentation

### 6.77.3.1 ColSumsImpl()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::ColSumsImpl (
    const MatrixBase< RTYPE, NA, T > & ref_ ) [inline]
```

Definition at line 420 of file rowSums.h.

## 6.77.4 Member Function Documentation

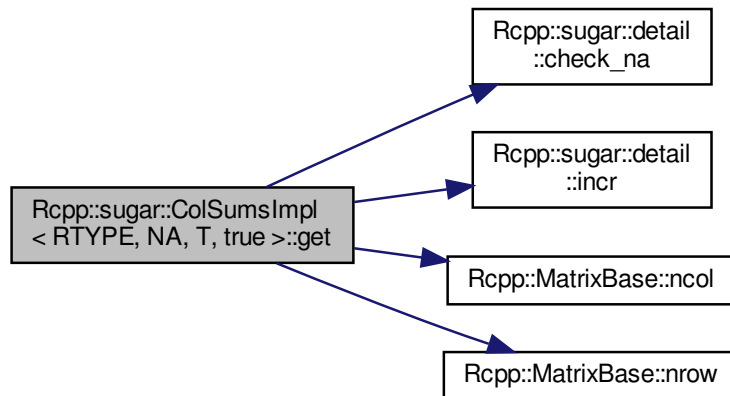
### 6.77.4.1 get()

```
template<int RTYPE, bool NA, typename T >
return_vector Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get ( ) const [inline]
```

Definition at line 424 of file rowSums.h.

References `Rcpp::sugar::detail::check_na()`, `Rcpp::sugar::detail::incr()`, `Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol()`, `Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow()`, and `Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA_RM >::ref`.

Here is the call graph for this function:



## 6.77.5 Member Data Documentation

### 6.77.5.1 ref

```
template<int RTYPE, bool NA, typename T >
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::ref [private]
```

Definition at line 413 of file rowSums.h.

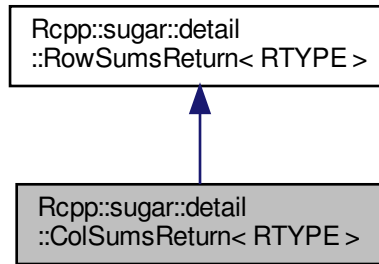
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/rowSums.h`

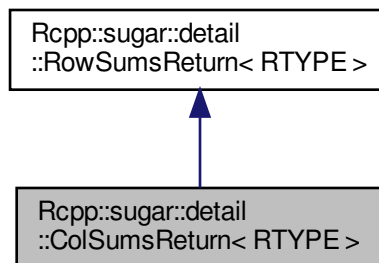
## 6.78 Rcpp::sugar::detail::ColSumsReturn< RTYPE > Struct Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::detail::ColSumsReturn< RTYPE >:



Collaboration diagram for Rcpp::sugar::detail::ColSumsReturn< RTYPE >:



### Additional Inherited Members

#### 6.78.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::sugar::detail::ColSumsReturn< RTYPE >
```

Definition at line 98 of file rowSums.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/rowSums.h](#)

## 6.79 Rcpp::attributes::CommentState Class Reference

### Public Member Functions

- [CommentState](#) ()
- bool [inComment](#) () const
- void [submitLine](#) (const std::string &line)
- void [reset](#) ()

### Private Member Functions

- [CommentState](#) (const [CommentState](#) &)
- [CommentState](#) & [operator=](#) (const [CommentState](#) &)

### Private Attributes

- bool [inComment\\_](#)

### 6.79.1 Detailed Description

Definition at line 470 of file attributes.cpp.

### 6.79.2 Constructor & Destructor Documentation

#### 6.79.2.1 CommentState() [1/2]

```
Rcpp::attributes::CommentState::CommentState ( ) [inline]
```

Definition at line 472 of file attributes.cpp.

#### 6.79.2.2 CommentState() [2/2]

```
Rcpp::attributes::CommentState::CommentState (
    const CommentState & ) [private]
```

## 6.79.3 Member Function Documentation

### 6.79.3.1 inComment()

```
bool Rcpp::attributes::CommentState::inComment ( ) const [inline]
```

Definition at line 478 of file attributes.cpp.

References inComment\_.

### 6.79.3.2 operator=()

```
CommentState& Rcpp::attributes::CommentState::operator= (
    const CommentState & ) [private]
```

### 6.79.3.3 reset()

```
void Rcpp::attributes::CommentState::reset ( ) [inline]
```

Definition at line 480 of file attributes.cpp.

References inComment\_.

### 6.79.3.4 submitLine()

```
void Rcpp::attributes::CommentState::submitLine (
    const std::string & line )
```

## 6.79.4 Member Data Documentation



### 6.79.4.1 inComment\_

```
bool Rcpp::attributes::CommentState::inComment_ [private]
```

Definition at line 482 of file attributes.cpp.

Referenced by inComment(), and reset().

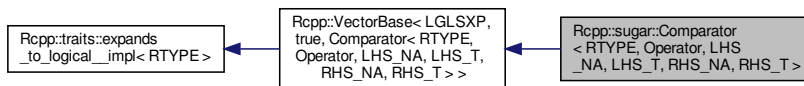
The documentation for this class was generated from the following file:

- [src/attributes.cpp](#)

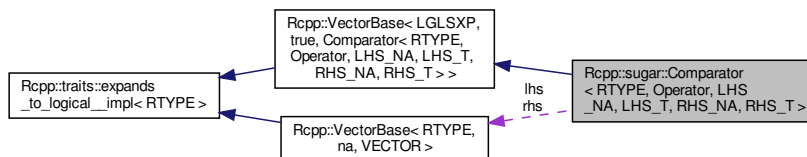
## 6.80 Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <Comparator.h>
```

Inheritance diagram for Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase< RTYPE, LHS\\_NA, LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase< RTYPE, RHS\\_NA, RHS\\_T >](#) [RHS\\_TYPE](#)
- typedef [traits::storage\\_type< RTYPE >::type](#) [STORAGE](#)

## Public Member Functions

- [Comparator](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- int [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_TYPE](#) & lhs
- const [RHS\\_TYPE](#) & rhs
- Operator [op](#)

### 6.80.1 Detailed Description

```
template<int RTYPE, typename Operator, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 29 of file Comparator.h.

### 6.80.2 Member Typedef Documentation

#### 6.80.2.1 LHS\_TYPE

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
typedef Rcpp::VectorBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 33 of file Comparator.h.

#### 6.80.2.2 RHS\_TYPE

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 34 of file Comparator.h.

### 6.80.2.3 STORAGE

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_←
_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 35 of file Comparator.h.

## 6.80.3 Constructor & Destructor Documentation

### 6.80.3.1 Comparator()

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::Comparator (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 37 of file Comparator.h.

## 6.80.4 Member Function Documentation

### 6.80.4.1 operator[]()

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
int Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 40 of file Comparator.h.

References Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, Rcpp::sugar::←  
Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::op, and Rcpp::sugar::Comparator< RTYPE,  
Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

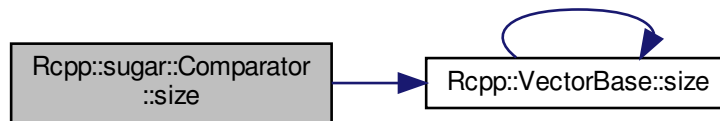
### 6.80.4.2 size()

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
R_xlen_t Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 48 of file Comparator.h.

References Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



## 6.80.5 Member Data Documentation

### 6.80.5.1 lhs

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T
>
const LHS_TYPE& Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 51 of file Comparator.h.

Referenced by Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, false, RHS\_T >::operator[](), Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, RHS\_NA, RHS\_T >::size(), and Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, false, RHS\_T >::size().

### 6.80.5.2 op

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T  
>  
Operator Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::op [private]
```

Definition at line 53 of file Comparator.h.

Referenced by Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, RHS\_NA, RHS\_T >::operator[](), and Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, false, RHS\_T >::operator[]().

### 6.80.5.3 rhs

```
template<int RTYPE, typename Operator , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T  
>  
const RHS_TYPE& Rcpp::sugar::Comparator< RTYPE, Operator, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs  
[private]
```

Definition at line 52 of file Comparator.h.

Referenced by Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, RHS\_NA, RHS\_T >::operator[](), and Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, false, RHS\_T >::operator[]().

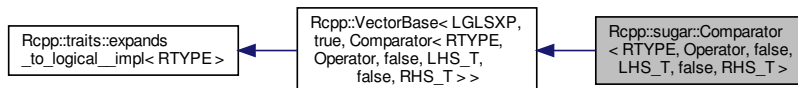
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/Comparator.h

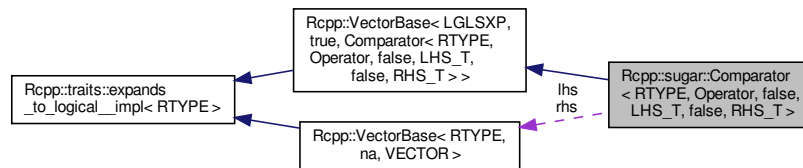
## 6.81 Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <Comparator.h>
```

Inheritance diagram for Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, false, RHS\_T >:



Collaboration diagram for `Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >`:



## Public Types

- typedef `Rcpp::VectorBase< RTYPE, false, LHS_T >` `LHS_TYPE`
- typedef `Rcpp::VectorBase< RTYPE, false, RHS_T >` `RHS_TYPE`
- typedef `traits::storage_type< RTYPE >::type` `STORAGE`

## Public Member Functions

- `Comparator` (const `LHS_TYPE` &lhs\_, const `RHS_TYPE` &rhs\_)
- int `operator[]` (`R_xlen_t` i) const
- `R_xlen_t` `size` () const

## Private Attributes

- const `LHS_TYPE` &lhs
- const `RHS_TYPE` &rhs
- Operator `op`

### 6.81.1 Detailed Description

```
template<int RTYPE, typename Operator, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >
```

Definition at line 88 of file `Comparator.h`.

### 6.81.2 Member Typedef Documentation

### 6.81.2.1 LHS\_TYPE

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Comparator< RTYPE, Operator, false,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 92 of file Comparator.h.

### 6.81.2.2 RHS\_TYPE

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Comparator< RTYPE, Operator, false,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 93 of file Comparator.h.

### 6.81.2.3 STORAGE

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T,
false, RHS_T >::STORAGE
```

Definition at line 94 of file Comparator.h.

## 6.81.3 Constructor & Destructor Documentation

### 6.81.3.1 Comparator()

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::Comparator (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 96 of file Comparator.h.

## 6.81.4 Member Function Documentation

### 6.81.4.1 operator[]()

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
int Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 99 of file Comparator.h.

References Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, Rcpp::sugar::←  
Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::op, and Rcpp::sugar::Comparator< RTYPE,  
Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

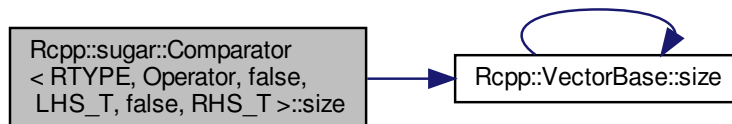
### 6.81.4.2 size()

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 103 of file Comparator.h.

References Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::←  
VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



## 6.81.5 Member Data Documentation

### 6.81.5.1 lhs

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::lhs [private]
```

Definition at line 106 of file Comparator.h.



## 6.82 Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

### 6.81.5.2 op

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >  
Operator Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::op [private]
```

Definition at line 108 of file Comparator.h.

### 6.81.5.3 rhs

```
template<int RTYPE, typename Operator , typename LHS_T , typename RHS_T >  
const RHS_TYPE& Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, false, RHS_T >::rhs [private]
```

Definition at line 107 of file Comparator.h.

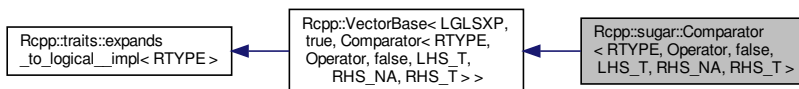
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/Comparator.h](#)

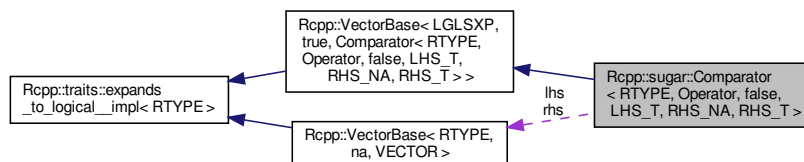
## 6.82 Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <Comparator.h>
```

Inheritance diagram for Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, RHS\_NA, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)

## Public Member Functions

- [Comparator](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- int [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_TYPE](#) &lhs
- const [RHS\\_TYPE](#) &rhs
- Operator [op](#)

### 6.82.1 Detailed Description

```
template<int RTYPE, typename Operator, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >
```

Definition at line 60 of file Comparator.h.

### 6.82.2 Member Typedef Documentation

#### 6.82.2.1 LHS\_TYPE

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Comparator< RTYPE, Operator, false,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 64 of file Comparator.h.

### 6.82.2.2 RHS\_TYPE

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Comparator< RTYPE, Operator, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 65 of file Comparator.h.

### 6.82.2.3 STORAGE

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 66 of file Comparator.h.

## 6.82.3 Constructor & Destructor Documentation

### 6.82.3.1 Comparator()

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::Comparator (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 68 of file Comparator.h.

## 6.82.4 Member Function Documentation

### 6.82.4.1 operator[]()

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
int Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 71 of file Comparator.h.

References Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::op, and Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

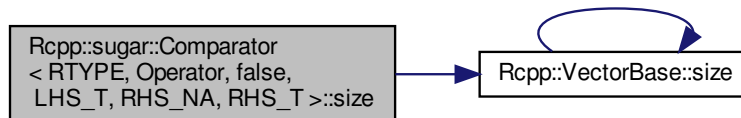
### 6.82.4.2 size()

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 77 of file Comparator.h.

References Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



## 6.82.5 Member Data Documentation

### 6.82.5.1 lhs

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 80 of file Comparator.h.

### 6.82.5.2 op

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
Operator Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::op [private]
```

Definition at line 82 of file Comparator.h.

6.82.5.3 rhs

```
template<int RTYPE, typename Operator , typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 81 of file Comparator.h.

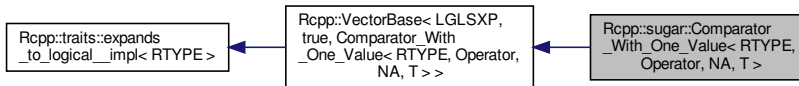
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/Comparator.h

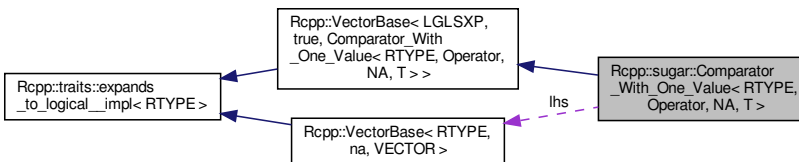
6.83 Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T > Class Template Reference

```
#include <Comparator_With_One_Value.h>
```

Inheritance diagram for Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >:



Collaboration diagram for Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, NA, T > VEC\_TYPE
- typedef traits::storage\_type< RTYPE >::type STORAGE
- typedef int(Comparator\_With\_One\_Value::\* METHOD) (int) const

## Public Member Functions

- [Comparator\\_With\\_One\\_Value](#) (const [VEC\\_TYPE](#) &lhs\_, [STORAGE](#) rhs\_)
- int [operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t\\_size](#) () const

## Private Member Functions

- int [rhs\\_is\\_na](#) (int i) const
- int [rhs\\_is\\_not\\_na](#) (int i) const

## Private Attributes

- const [VEC\\_TYPE](#) & lhs
- [STORAGE](#) rhs
- [METHOD](#) m
- Operator [op](#)

### 6.83.1 Detailed Description

```
template<int RTYPE, typename Operator, bool NA, typename T>
class Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >
```

Definition at line 29 of file `Comparator_With_One_Value.h`.

### 6.83.2 Member Typedef Documentation

#### 6.83.2.1 METHOD

```
template<int RTYPE, typename Operator , bool NA, typename T >
typedef int(Comparator_With_One_Value::* Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator,
NA, T >::METHOD) (int) const
```

Definition at line 33 of file `Comparator_With_One_Value.h`.

### 6.83.2.2 STORAGE

```
template<int RTYPE, typename Operator , bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator,
NA, T >::STORAGE
```

Definition at line 32 of file Comparator\_With\_One\_Value.h.

### 6.83.2.3 VEC\_TYPE

```
template<int RTYPE, typename Operator , bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA,
T >::VEC_TYPE
```

Definition at line 31 of file Comparator\_With\_One\_Value.h.

## 6.83.3 Constructor & Destructor Documentation

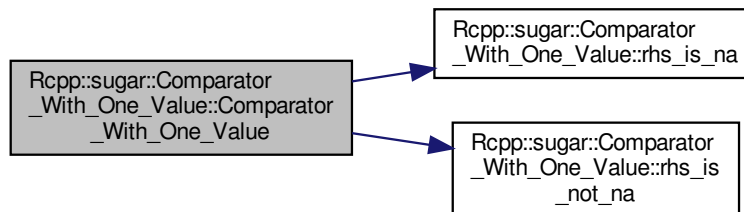
### 6.83.3.1 Comparator\_With\_One\_Value()

```
template<int RTYPE, typename Operator , bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::Comparator_With_One_Value (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 35 of file Comparator\_With\_One\_Value.h.

References Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::m, Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs, Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs\_is\_na(), and Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs\_is\_not\_na().

Here is the call graph for this function:



## 6.83.4 Member Function Documentation

### 6.83.4.1 operator[]()

```
template<int RTYPE, typename Operator , bool NA, typename T >
int Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 44 of file Comparator\_With\_One\_Value.h.

References Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::m.

### 6.83.4.2 rhs\_is\_na()

```
template<int RTYPE, typename Operator , bool NA, typename T >
int Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs_is_na (
    int i ) const [inline], [private]
```

Definition at line 56 of file Comparator\_With\_One\_Value.h.

References Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs.

Referenced by Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::Comparator\_With\_One\_↔Value(), and Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T >::Comparator\_With\_One\_↔Value().

### 6.83.4.3 rhs\_is\_not\_na()

```
template<int RTYPE, typename Operator , bool NA, typename T >
int Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs_is_not_na (
    int i ) const [inline], [private]
```

Definition at line 57 of file Comparator\_With\_One\_Value.h.

References Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::lhs, Rcpp::sugar::Comparator\_↔\_With\_One\_Value< RTYPE, Operator, NA, T >::op, and Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs.

Referenced by Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::Comparator\_With\_One\_↔Value(), and Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T >::Comparator\_With\_One\_↔Value().



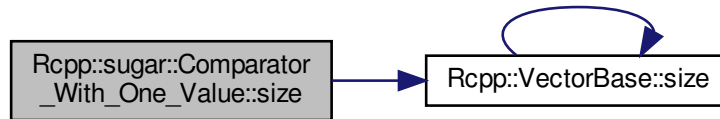
#### 6.83.4.4 size()

```
template<int RTYPE, typename Operator , bool NA, typename T >
R_xlen_t Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::size ( ) const [inline]
```

Definition at line 48 of file Comparator\_With\_One\_Value.h.

References Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



### 6.83.5 Member Data Documentation

#### 6.83.5.1 lhs

```
template<int RTYPE, typename Operator , bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::lhs [private]
```

Definition at line 51 of file Comparator\_With\_One\_Value.h.

Referenced by Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs\_is\_not\_na(), Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T >::rhs\_is\_not\_na(), Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::size(), and Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T >::size().

#### 6.83.5.2 m

```
template<int RTYPE, typename Operator , bool NA, typename T >
METHOD Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::m [private]
```

Definition at line 53 of file Comparator\_With\_One\_Value.h.

Referenced by Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::Comparator\_With\_One\_Value(), Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T >::Comparator\_With\_One\_Value(), Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::operator[](), and Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T >::operator[]().

### 6.83.5.3 op

```
template<int RTYPE, typename Operator , bool NA, typename T >
Operator Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::op [private]
```

Definition at line 54 of file Comparator\_With\_One\_Value.h.

Referenced by Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs\_is\_not\_na(), and Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T >::rhs\_is\_not\_na().

### 6.83.5.4 rhs

```
template<int RTYPE, typename Operator , bool NA, typename T >
STORAGE Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, NA, T >::rhs [private]
```

Definition at line 52 of file Comparator\_With\_One\_Value.h.

Referenced by Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::Comparator\_With\_One\_Value(), Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T >::Comparator\_With\_One\_Value(), Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs\_is\_na(), Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T >::rhs\_is\_na(), Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs\_is\_not\_na(), and Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T >::rhs\_is\_not\_na().

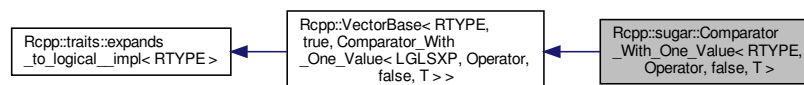
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/Comparator\_With\_One\_Value.h

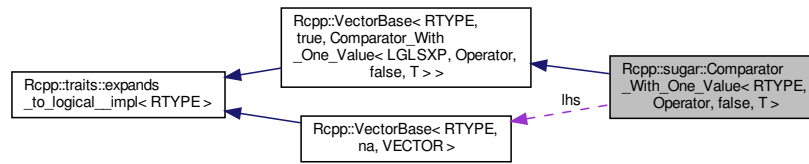
## 6.84 Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T > Class Template Reference

```
#include <Comparator_With_One_Value.h>
```

Inheritance diagram for Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T >:



Collaboration diagram for Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, false, T > [VEC\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef int([Comparator\\_With\\_One\\_Value](#)::\* [METHOD](#)) (int) const

## Public Member Functions

- [Comparator\\_With\\_One\\_Value](#) (const [VEC\\_TYPE](#) &lhs\_, [STORAGE](#) rhs\_)
- int [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Member Functions

- int [rhs\\_is\\_na](#) (int i) const
- int [rhs\\_is\\_not\\_na](#) (int i) const

## Private Attributes

- const [VEC\\_TYPE](#) & [lhs](#)
- [STORAGE](#) [rhs](#)
- [METHOD](#) [m](#)
- Operator [op](#)

### 6.84.1 Detailed Description

```

template<int RTYPE, typename Operator, typename T>
class Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >
  
```

Definition at line 66 of file [Comparator\\_With\\_One\\_Value.h](#).

## 6.84.2 Member Typedef Documentation

### 6.84.2.1 METHOD

```
template<int RTYPE, typename Operator , typename T >
typedef int(Comparator_With_One_Value::* Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator,
false, T >::METHOD) (int) const
```

Definition at line 72 of file Comparator\_With\_One\_Value.h.

### 6.84.2.2 STORAGE

```
template<int RTYPE, typename Operator , typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator,
false, T >::STORAGE
```

Definition at line 71 of file Comparator\_With\_One\_Value.h.

### 6.84.2.3 VEC\_TYPE

```
template<int RTYPE, typename Operator , typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator,
false, T >::VEC_TYPE
```

Definition at line 70 of file Comparator\_With\_One\_Value.h.

## 6.84.3 Constructor & Destructor Documentation

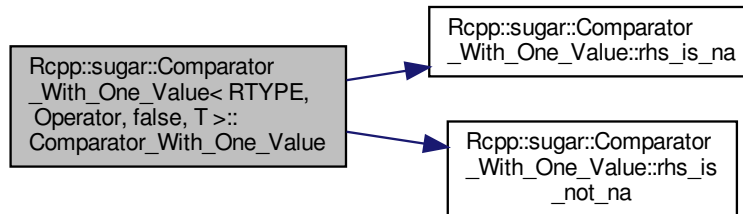
### 6.84.3.1 Comparator\_With\_One\_Value()

```
template<int RTYPE, typename Operator , typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::Comparator_With_One_Value (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 74 of file Comparator\_With\_One\_Value.h.

References Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::m, Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs, Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs\_is\_na(), and Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs\_is\_not\_na().

Here is the call graph for this function:



## 6.84.4 Member Function Documentation

### 6.84.4.1 operator[]()

```
template<int RTYPE, typename Operator , typename T >
int Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 83 of file Comparator\_With\_One\_Value.h.

References Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::m.

#### 6.84.4.2 rhs\_is\_na()

```
template<int RTYPE, typename Operator , typename T >
int Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::rhs_is_na (
    int i ) const [inline], [private]
```

Definition at line 95 of file Comparator\_With\_One\_Value.h.

References Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs.

#### 6.84.4.3 rhs\_is\_not\_na()

```
template<int RTYPE, typename Operator , typename T >
int Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::rhs_is_not_na (
    int i ) const [inline], [private]
```

Definition at line 96 of file Comparator\_With\_One\_Value.h.

References Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::lhs, Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::op, and Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::rhs.

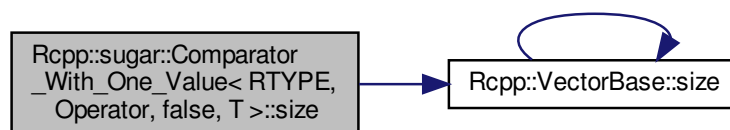
#### 6.84.4.4 size()

```
template<int RTYPE, typename Operator , typename T >
R_xlen_t Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::size ( ) const
[inline]
```

Definition at line 87 of file Comparator\_With\_One\_Value.h.

References Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



## 6.84.5 Member Data Documentation

### 6.84.5.1 lhs

```
template<int RTYPE, typename Operator , typename T >
const VEC_TYPE& Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::lhs [private]
```

Definition at line 90 of file Comparator\_With\_One\_Value.h.

### 6.84.5.2 m

```
template<int RTYPE, typename Operator , typename T >
METHOD Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::m [private]
```

Definition at line 92 of file Comparator\_With\_One\_Value.h.

### 6.84.5.3 op

```
template<int RTYPE, typename Operator , typename T >
Operator Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::op [private]
```

Definition at line 93 of file Comparator\_With\_One\_Value.h.

### 6.84.5.4 rhs

```
template<int RTYPE, typename Operator , typename T >
STORAGE Rcpp::sugar::Comparator_With_One_Value< RTYPE, Operator, false, T >::rhs [private]
```

Definition at line 91 of file Comparator\_With\_One\_Value.h.

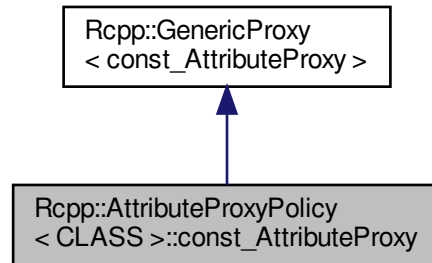
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/Comparator\\_With\\_One\\_Value.h](#)

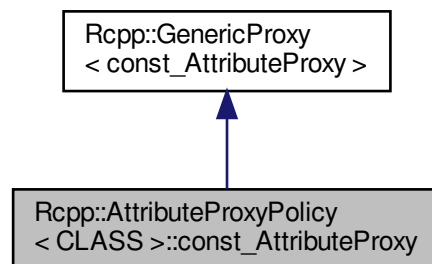
## 6.85 Rcpp::AttributeProxyPolicy< CLASS >::const\_AttributeProxy Class Reference

```
#include <AttributeProxy.h>
```

Inheritance diagram for Rcpp::AttributeProxyPolicy< CLASS >::const\_AttributeProxy:



Collaboration diagram for Rcpp::AttributeProxyPolicy< CLASS >::const\_AttributeProxy:



### Public Member Functions

- [const\\_AttributeProxy](#) (const CLASS &v, const std::string &name)
- `template<typename T >`  
[operator T](#) () const
- [operator SEXP](#) () const



## Private Member Functions

- SEXP [get](#) () const

## Private Attributes

- const CLASS & [parent](#)
- SEXP [attr\\_name](#)

### 6.85.1 Detailed Description

```
template<typename CLASS>  
class Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy
```

Definition at line 56 of file AttributeProxy.h.

### 6.85.2 Constructor & Destructor Documentation

#### 6.85.2.1 const\_AttributeProxy()

```
template<typename CLASS >  
Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::const_AttributeProxy (  
    const CLASS & v,  
    const std::string & name ) [inline]
```

Definition at line 58 of file AttributeProxy.h.

### 6.85.3 Member Function Documentation

#### 6.85.3.1 get()

```
template<typename CLASS >  
SEXP Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::get ( ) const [inline], [private]
```

Definition at line 68 of file AttributeProxy.h.

References [Rcpp::AttributeProxyPolicy< CLASS >::const\\_AttributeProxy::attr\\_name](#), and [Rcpp::AttributeProxyPolicy< CLASS >::const\\_AttributeProxy::parent](#).

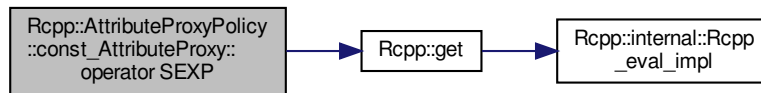
### 6.85.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::operator SEXP [inline]
```

Definition at line 55 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



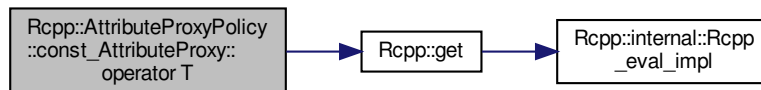
### 6.85.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::operator T
```

Definition at line 50 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



## 6.85.4 Member Data Documentation

### 6.85.4.1 attr\_name

```
template<typename CLASS >
SEXP Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::attr_name [private]
```

Definition at line 66 of file AttributeProxy.h.

Referenced by Rcpp::AttributeProxyPolicy< CLASS >::const\_AttributeProxy::get().

### 6.85.4.2 parent

```
template<typename CLASS >
const CLASS& Rcpp::AttributeProxyPolicy< CLASS >::const_AttributeProxy::parent [private]
```

Definition at line 65 of file AttributeProxy.h.

Referenced by Rcpp::AttributeProxyPolicy< CLASS >::const\_AttributeProxy::get().

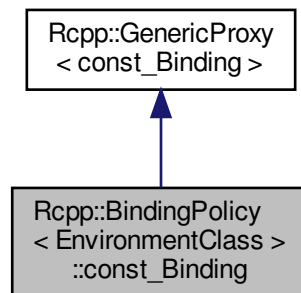
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/AttributeProxy.h
- inst/include/Rcpp/api/meat/proxy.h

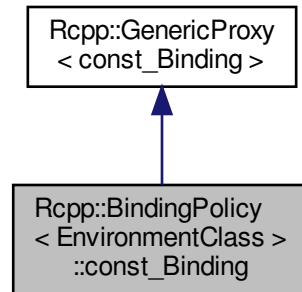
## 6.86 Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding Class Reference

```
#include <Binding.h>
```

Inheritance diagram for Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding:



Collaboration diagram for Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding:



## Public Member Functions

- `const_Binding` (const EnvironmentClass &env\_, const std::string &name\_)
- bool `active` () const
- bool `locked` () const
- bool `exists` () const
- template<typename T >  
operator T () const

## Private Member Functions

- SEXP `get` () const

## Private Attributes

- const EnvironmentClass & `env`
- std::string `name`

### 6.86.1 Detailed Description

```

template<typename EnvironmentClass>
class Rcpp::BindingPolicy< EnvironmentClass >::const_Binding
  
```

Definition at line 72 of file Binding.h.

## 6.86.2 Constructor & Destructor Documentation

### 6.86.2.1 const\_Binding()

```
template<typename EnvironmentClass >  
Rcpp::BindingPolicy< EnvironmentClass >::const_Binding (   
    const EnvironmentClass & env_,  
    const std::string & name_ ) [inline]
```

Definition at line 74 of file Binding.h.

## 6.86.3 Member Function Documentation

### 6.86.3.1 active()

```
template<typename EnvironmentClass >  
bool Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::active ( ) const [inline]
```

Definition at line 77 of file Binding.h.

References [Rcpp::BindingPolicy< EnvironmentClass >::const\\_Binding::env](#), and [Rcpp::BindingPolicy< EnvironmentClass >::const\\_Binding::name](#).

### 6.86.3.2 exists()

```
template<typename EnvironmentClass >  
bool Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::exists ( ) const [inline]
```

Definition at line 83 of file Binding.h.

References [Rcpp::BindingPolicy< EnvironmentClass >::const\\_Binding::env](#), and [Rcpp::BindingPolicy< EnvironmentClass >::const\\_Binding::name](#).

### 6.86.3.3 get()

```
template<typename EnvironmentClass >
SEXPRcpp::BindingPolicy< EnvironmentClass >::const_Binding::get ( ) const [inline], [private]
```

Definition at line 90 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding::name.

### 6.86.3.4 locked()

```
template<typename EnvironmentClass >
bool Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::locked ( ) const [inline]
```

Definition at line 80 of file Binding.h.

References Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding::env, and Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding::name.

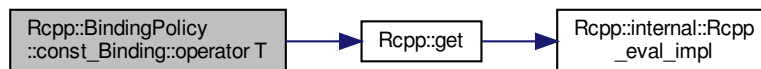
### 6.86.3.5 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::BindingPolicy< CLASS >::const_Binding::operator T
```

Definition at line 143 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



## 6.86.4 Member Data Documentation

### 6.86.4.1 env

```
template<typename EnvironmentClass >
const EnvironmentClass& Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::env [private]
```

Definition at line 94 of file Binding.h.

Referenced by Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding::active(), Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding::exists(), Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding::get(), and Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding::locked().

### 6.86.4.2 name

```
template<typename EnvironmentClass >
std::string Rcpp::BindingPolicy< EnvironmentClass >::const_Binding::name [private]
```

Definition at line 95 of file Binding.h.

Referenced by Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding::active(), Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding::exists(), Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding::get(), and Rcpp::BindingPolicy< EnvironmentClass >::const\_Binding::locked().

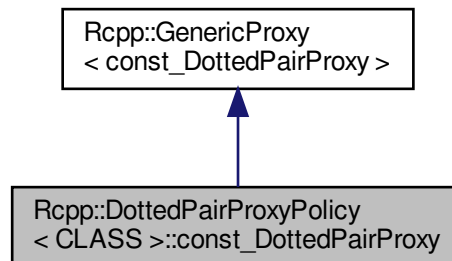
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/Binding.h
- inst/include/Rcpp/api/meat/proxy.h

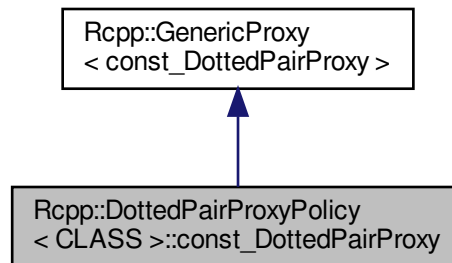
## 6.87 Rcpp::DottedPairProxyPolicy< CLASS >::const\_DottedPairProxy Class Reference

```
#include <DottedPairProxy.h>
```

Inheritance diagram for Rcpp::DottedPairProxyPolicy< CLASS >::const\_DottedPairProxy:



Collaboration diagram for Rcpp::DottedPairProxyPolicy< CLASS >::const\_DottedPairProxy:



## Public Member Functions

- [const\\_DottedPairProxy](#) (const CLASS &v, int index\_)
- `template<typename T >`  
[operator T](#) () const
- SEXP [get](#) () const
- [operator SEXP](#) () const

## Private Attributes

- SEXP [node](#)

### 6.87.1 Detailed Description

```
template<typename CLASS>
class Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy
```

Definition at line 77 of file DottedPairProxy.h.

### 6.87.2 Constructor & Destructor Documentation



### 6.87.2.1 const\_DottedPairProxy()

```
template<typename CLASS >  
Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::const_DottedPairProxy (  
    const CLASS & v,  
    int index_ ) [inline]
```

Definition at line 79 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::const\_DottedPairProxy::node.

## 6.87.3 Member Function Documentation

### 6.87.3.1 get()

```
template<typename CLASS >  
SEXP Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::get ( ) const [inline]
```

Definition at line 93 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::const\_DottedPairProxy::node.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::const\_DottedPairProxy::operator SEXP().

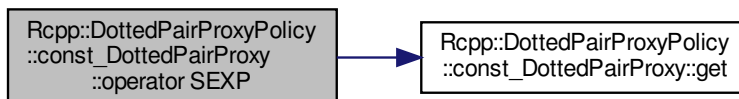
### 6.87.3.2 operator SEXP()

```
template<typename CLASS >  
Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::operator SEXP ( ) const [inline]
```

Definition at line 96 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::const\_DottedPairProxy::get().

Here is the call graph for this function:



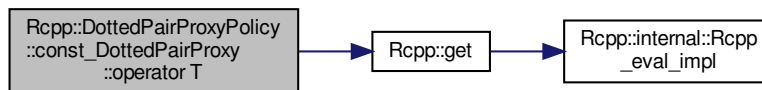
### 6.87.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::operator T
```

Definition at line 171 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



## 6.87.4 Member Data Documentation

### 6.87.4.1 node

```
template<typename CLASS >
SEXP Rcpp::DottedPairProxyPolicy< CLASS >::const_DottedPairProxy::node [private]
```

Definition at line 101 of file DottedPairProxy.h.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::const\_DottedPairProxy::const\_DottedPairProxy(), and Rcpp::DottedPairProxyPolicy< CLASS >::const\_DottedPairProxy::get().

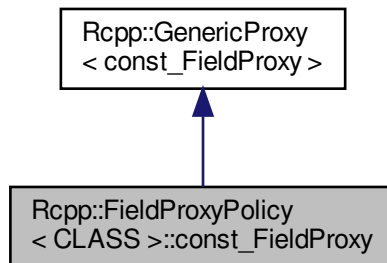
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/DottedPairProxy.h
- inst/include/Rcpp/api/meat/proxy.h

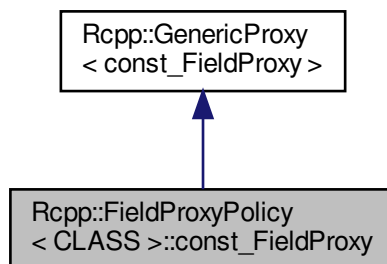
## 6.88 Rcpp::FieldProxyPolicy< CLASS >::const\_FieldProxy Class Reference

```
#include <FieldProxy.h>
```

Inheritance diagram for Rcpp::FieldProxyPolicy< CLASS >::const\_FieldProxy:



Collaboration diagram for Rcpp::FieldProxyPolicy< CLASS >::const\_FieldProxy:



### Public Member Functions

- [const\\_FieldProxy](#) (const CLASS &v, const std::string &name)
- [template<typename T > operator T \(\) const](#)
- [operator SEXP \(\) const](#)

## Private Member Functions

- SEXP [get](#) () const

## Private Attributes

- const CLASS & [parent](#)
- const std::string & [field\\_name](#)

### 6.88.1 Detailed Description

```
template<typename CLASS>  
class Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy
```

Definition at line 56 of file FieldProxy.h.

### 6.88.2 Constructor & Destructor Documentation

#### 6.88.2.1 const\_FieldProxy()

```
template<typename CLASS >  
Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::const_FieldProxy (  
    const CLASS & v,  
    const std::string & name ) [inline]
```

Definition at line 58 of file FieldProxy.h.

### 6.88.3 Member Function Documentation



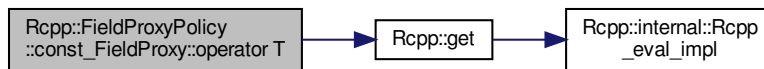
### 6.88.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::operator T
```

Definition at line 199 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



## 6.88.4 Member Data Documentation

### 6.88.4.1 field\_name

```
template<typename CLASS >
const std::string& Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::field_name [private]
```

Definition at line 68 of file FieldProxy.h.

Referenced by Rcpp::FieldProxyPolicy< CLASS >::const\_FieldProxy::get().

### 6.88.4.2 parent

```
template<typename CLASS >
const CLASS& Rcpp::FieldProxyPolicy< CLASS >::const_FieldProxy::parent [private]
```

Definition at line 67 of file FieldProxy.h.

Referenced by Rcpp::FieldProxyPolicy< CLASS >::const\_FieldProxy::get().

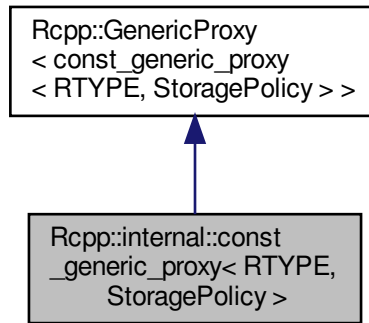
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/[FieldProxy.h](#)
- inst/include/Rcpp/api/meat/[proxy.h](#)

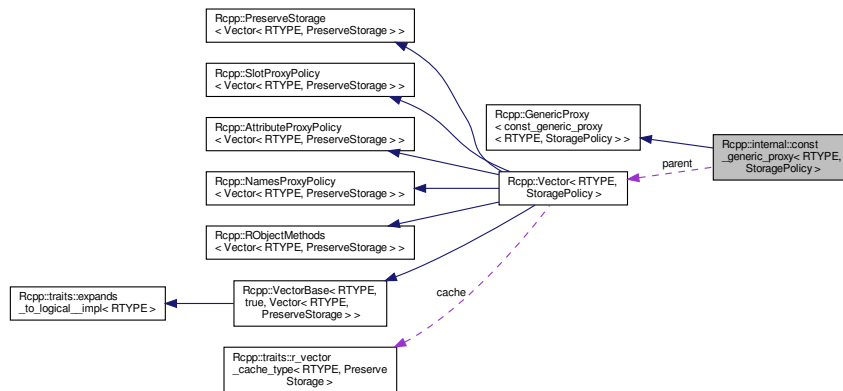
## 6.89 Rcpp::internal::const\_generic\_proxy< RTYPE, StoragePolicy > Class Template Reference

```
#include <const_generic_proxy.h>
```

Inheritance diagram for Rcpp::internal::const\_generic\_proxy< RTYPE, StoragePolicy >:



Collaboration diagram for Rcpp::internal::const\_generic\_proxy< RTYPE, StoragePolicy >:



### Public Types

- typedef `::Rcpp::Vector< RTYPE, StoragePolicy >` `VECTOR`

## Public Member Functions

- [const\\_generic\\_proxy](#) ()
- [const\\_generic\\_proxy](#) (const [const\\_generic\\_proxy](#) &other)
- [const\\_generic\\_proxy](#) (const [VECTOR](#) &v, [R\\_xlen\\_t](#) i)
- [operator SEXP](#) () const
- [template<typename U > operator U](#) () const
- [operator bool](#) () const
- [operator int](#) () const
- void [move](#) ([R\\_xlen\\_t](#) n)

## Public Attributes

- const [VECTOR](#) \* [parent](#)
- [R\\_xlen\\_t](#) [index](#)

## Private Member Functions

- [SEXP](#) [get](#) () const

### 6.89.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>
class Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >
```

Definition at line 27 of file `const_generic_proxy.h`.

### 6.89.2 Member Typedef Documentation

#### 6.89.2.1 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::VECTOR
```

Definition at line 29 of file `const_generic_proxy.h`.

### 6.89.3 Constructor & Destructor Documentation



### 6.89.3.1 const\_generic\_proxy() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::const_generic_proxy ( ) [inline]
```

Definition at line 31 of file const\_generic\_proxy.h.

### 6.89.3.2 const\_generic\_proxy() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::const_generic_proxy (
    const const_generic_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 33 of file const\_generic\_proxy.h.

### 6.89.3.3 const\_generic\_proxy() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::const_generic_proxy (
    const VECTOR & v,
    R_xlen_t i ) [inline]
```

Definition at line 36 of file const\_generic\_proxy.h.

## 6.89.4 Member Function Documentation

### 6.89.4.1 get()

```
template<int RTYPE, template< class > class StoragePolicy>
SEXP Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::get ( ) const [inline], [private]
```

Definition at line 57 of file const\_generic\_proxy.h.

References [Rcpp::internal::const\\_generic\\_proxy< RTYPE, StoragePolicy >::index](#), and [Rcpp::internal::const\\_↔generic\\_proxy< RTYPE, StoragePolicy >::parent](#).

Referenced by [Rcpp::internal::const\\_generic\\_proxy< RTYPE, StoragePolicy >::operator bool\(\)](#), [Rcpp::internal::const\\_↔generic\\_proxy< RTYPE, StoragePolicy >::operator int\(\)](#), [Rcpp::internal::const\\_generic\\_proxy< RTYPE, Storage\\_↔Policy >::operator SEXP\(\)](#), and [Rcpp::internal::const\\_generic\\_proxy< RTYPE, StoragePolicy >::operator U\(\)](#).

**6.89.4.2 move()**

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::move (
    R_xlen_t n) [inline]
```

Definition at line 50 of file const\_generic\_proxy.h.

References Rcpp::internal::const\_generic\_proxy< RTYPE, StoragePolicy >::index.

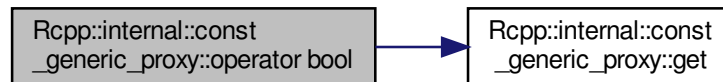
**6.89.4.3 operator bool()**

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::operator bool ( ) const [inline]
```

Definition at line 47 of file const\_generic\_proxy.h.

References Rcpp::internal::const\_generic\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:

**6.89.4.4 operator int()**

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::operator int ( ) const [inline]
```

Definition at line 48 of file const\_generic\_proxy.h.

References Rcpp::internal::const\_generic\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



#### 6.89.4.5 operator SEXP()

```
template<int RTYPE, template< class > class StoragePolicy>  
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const [inline]
```

Definition at line 38 of file const\_generic\_proxy.h.

References Rcpp::internal::const\_generic\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



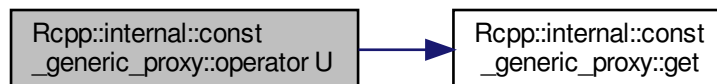
#### 6.89.4.6 operator U()

```
template<int RTYPE, template< class > class StoragePolicy>  
template<typename U >  
Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::operator U ( ) const [inline]
```

Definition at line 42 of file const\_generic\_proxy.h.

References Rcpp::internal::const\_generic\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



### 6.89.5 Member Data Documentation

### 6.89.5.1 index

```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::index
```

Definition at line 53 of file const\_generic\_proxy.h.

Referenced by Rcpp::internal::const\_generic\_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::const\_generic\_proxy< RTYPE, StoragePolicy >::move().

### 6.89.5.2 parent

```
template<int RTYPE, template< class > class StoragePolicy>
const VECTOR* Rcpp::internal::const_generic_proxy< RTYPE, StoragePolicy >::parent
```

Definition at line 52 of file const\_generic\_proxy.h.

Referenced by Rcpp::internal::const\_generic\_proxy< RTYPE, StoragePolicy >::get().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00\_forward\_proxy.h
- inst/include/Rcpp/vector/const\_generic\_proxy.h

## 6.90 Rcpp::MatrixRow< RTYPE >::const\_iter\_traits Struct Reference

```
#include <MatrixRow.h>
```

### Public Types

- typedef traits::r\_vector\_const\_iterator< RTYPE >::type vector\_iterator
- typedef int difference\_type
- typedef traits::r\_vector\_proxy< RTYPE >::type value\_type
- typedef value\_type reference
- typedef std::iterator\_traits< vector\_iterator >::pointer pointer

### 6.90.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::MatrixRow< RTYPE >::const_iter_traits
```

Definition at line 51 of file MatrixRow.h.

## 6.90.2 Member Typedef Documentation

### 6.90.2.1 difference\_type

```
template<int RTYPE>
typedef int Rcpp::MatrixRow< RTYPE >::const_iter_traits::difference_type
```

Definition at line 55 of file MatrixRow.h.

### 6.90.2.2 pointer

```
template<int RTYPE>
typedef std::iterator_traits<vector_iterator>::pointer Rcpp::MatrixRow< RTYPE >::const_iter_traits::pointer
```

Definition at line 58 of file MatrixRow.h.

### 6.90.2.3 reference

```
template<int RTYPE>
typedef value_type Rcpp::MatrixRow< RTYPE >::const_iter_traits::reference
```

Definition at line 57 of file MatrixRow.h.

### 6.90.2.4 value\_type

```
template<int RTYPE>
typedef traits::r_vector_proxy<RTYPE>::type Rcpp::MatrixRow< RTYPE >::const_iter_traits::value_type
```

Definition at line 56 of file MatrixRow.h.

### 6.90.2.5 vector\_iterator

```
template<int RTYPE>
typedef traits::r_vector_const_iterator<RTYPE>::type Rcpp::MatrixRow< RTYPE >::const_iter_traits::vector_iterator
```

Definition at line 53 of file MatrixRow.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/MatrixRow.h](#)

## 6.91 Rcpp::VectorBase< RTYPE, na, VECTOR >::const\_iter\_traits Struct Reference

```
#include <VectorBase.h>
```

### Public Types

- typedef [stored\\_type](#) [reference](#)
- typedef [stored\\_type](#) const \* [pointer](#)
- typedef R\_xlen\_t [difference\\_type](#)
- typedef const [stored\\_type](#) [value\\_type](#)
- typedef std::random\_access\_iterator\_tag [iterator\\_category](#)

### 6.91.1 Detailed Description

```
template<int RTYPE, bool na, typename VECTOR>
struct Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits
```

Definition at line 65 of file VectorBase.h.

### 6.91.2 Member Typedef Documentation

#### 6.91.2.1 difference\_type

```
template<int RTYPE, bool na, typename VECTOR >
typedef R_xlen_t Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits::difference_type
```

Definition at line 69 of file VectorBase.h.

### 6.91.2.2 iterator\_category

```
template<int RTYPE, bool na, typename VECTOR >  
typedef std::random_access_iterator_tag Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits::iterator_category
```

Definition at line 71 of file VectorBase.h.

### 6.91.2.3 pointer

```
template<int RTYPE, bool na, typename VECTOR >  
typedef stored_type const* Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits::pointer
```

Definition at line 68 of file VectorBase.h.

### 6.91.2.4 reference

```
template<int RTYPE, bool na, typename VECTOR >  
typedef stored_type Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits::reference
```

Definition at line 67 of file VectorBase.h.

### 6.91.2.5 value\_type

```
template<int RTYPE, bool na, typename VECTOR >  
typedef const stored_type Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iter_traits::value_type
```

Definition at line 70 of file VectorBase.h.

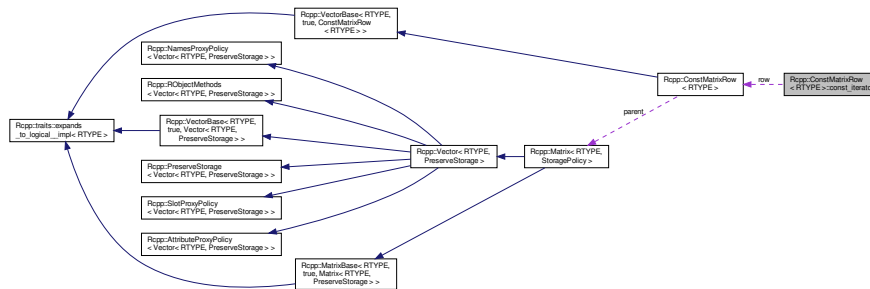
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/VectorBase.h](#)

## 6.92 Rcpp::ConstMatrixRow< RTYPE >::const\_iterator Class Reference

```
#include <MatrixRow.h>
```

Collaboration diagram for Rcpp::ConstMatrixRow< RTYPE >::const\_iterator:



### Public Types

- typedef `traits::r_vector_iterator< RTYPE >::type` `vector_iterator`
- typedef int `difference_type`
- typedef `traits::r_vector_const_proxy< RTYPE >::type` `value_type`
- typedef `traits::r_vector_const_proxy< RTYPE >::type` `reference`
- typedef `std::iterator_traits< vector_iterator >::pointer` `pointer`
- typedef `std::random_access_iterator_tag` `iterator_category`

### Public Member Functions

- `const_iterator` (`const const_iterator &other`)
- `const_iterator` (`const ConstMatrixRow &row_, int index_`)
- `const_iterator & operator++` ()
- `const_iterator operator++` (`int`)
- `const_iterator & operator--` ()
- `const_iterator operator--` (`int`)
- `const_iterator operator+` (`difference_type n`) `const`
- `const_iterator operator-` (`difference_type n`) `const`
- `difference_type operator-` (`const const_iterator &other`) `const`
- `const_iterator & operator+=` (`difference_type n`)
- `const_iterator & operator-=` (`difference_type n`)
- `const reference operator*` ()
- `const pointer operator->` ()
- `bool operator==` (`const const_iterator &other`)
- `bool operator!=` (`const const_iterator &other`)
- `bool operator<` (`const const_iterator &other`)
- `bool operator>` (`const const_iterator &other`)
- `bool operator<=` (`const const_iterator &other`)
- `bool operator>=` (`const const_iterator &other`)
- `const reference operator[]` (`int i`) `const`
- `difference_type operator-` (`const const_iterator &other`)



## Private Attributes

- const [ConstMatrixRow](#) & [row](#)
- int [index](#)

### 6.92.1 Detailed Description

```
template<int RTYPE>
class Rcpp::ConstMatrixRow< RTYPE >::const_iterator
```

Definition at line 222 of file MatrixRow.h.

### 6.92.2 Member Typedef Documentation

#### 6.92.2.1 difference\_type

```
template<int RTYPE>
typedef int Rcpp::ConstMatrixRow< RTYPE >::const_iterator::difference_type
```

Definition at line 226 of file MatrixRow.h.

#### 6.92.2.2 iterator\_category

```
template<int RTYPE>
typedef std::random_access_iterator_tag Rcpp::ConstMatrixRow< RTYPE >::const_iterator::↔
_iterator_category
```

Definition at line 231 of file MatrixRow.h.

#### 6.92.2.3 pointer

```
template<int RTYPE>
typedef std::iterator_traits<vector\_iterator>::pointer Rcpp::ConstMatrixRow< RTYPE >::const_↔
_iterator::pointer
```

Definition at line 229 of file MatrixRow.h.

#### 6.92.2.4 reference

```
template<int RTYPE>
typedef traits::r_vector_const_proxy<RTYPE>::type Rcpp::ConstMatrixRow< RTYPE >::const_iterator↔
::reference
```

Definition at line 228 of file MatrixRow.h.

#### 6.92.2.5 value\_type

```
template<int RTYPE>
typedef traits::r_vector_const_proxy<RTYPE>::type Rcpp::ConstMatrixRow< RTYPE >::const_iterator↔
::value_type
```

Definition at line 227 of file MatrixRow.h.

#### 6.92.2.6 vector\_iterator

```
template<int RTYPE>
typedef traits::r_vector_iterator<RTYPE>::type Rcpp::ConstMatrixRow< RTYPE >::const_iterator↔
::vector_iterator
```

Definition at line 224 of file MatrixRow.h.

### 6.92.3 Constructor & Destructor Documentation

#### 6.92.3.1 const\_iterator() [1/2]

```
template<int RTYPE>
Rcpp::ConstMatrixRow< RTYPE >::const_iterator::const_iterator (
    const const_iterator & other ) [inline]
```

Definition at line 233 of file MatrixRow.h.

### 6.92.3.2 const\_iterator() [2/2]

```
template<int RTYPE>
Rcpp::ConstMatrixRow< RTYPE >::const_iterator::const_iterator (
    const ConstMatrixRow & row_,
    int index_ ) [inline]
```

Definition at line 234 of file MatrixRow.h.

## 6.92.4 Member Function Documentation

### 6.92.4.1 operator"!=(())

```
template<int RTYPE>
bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator!= (
    const const_iterator & other ) [inline]
```

Definition at line 271 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

### 6.92.4.2 operator\*()

```
template<int RTYPE>
const reference Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator* ( ) [inline]
```

Definition at line 263 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index, and Rcpp::ConstMatrixRow< RTYPE >↔::const\_iterator::row.

### 6.92.4.3 operator+()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator+ (
    difference_type n ) const [inline]
```

Definition at line 256 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index, and Rcpp::ConstMatrixRow< RTYPE >↔::const\_iterator::row.

#### 6.92.4.4 operator++() [1/2]

```
template<int RTYPE>
const_iterator& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator++ ( ) [inline]
```

Definition at line 236 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

#### 6.92.4.5 operator++() [2/2]

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator++ (
    int ) [inline]
```

Definition at line 240 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

#### 6.92.4.6 operator+=()

```
template<int RTYPE>
const_iterator& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator+= (
    difference_type n ) [inline]
```

Definition at line 260 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

#### 6.92.4.7 operator-() [1/3]

```
template<int RTYPE>
difference_type Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator- (
    const const_iterator & other ) [inline]
```

Definition at line 281 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

#### 6.92.4.8 operator-() [2/3]

```
template<int RTYPE>
difference_type Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator- (
    const const_iterator & other ) const [inline]
```

Definition at line 258 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

#### 6.92.4.9 operator-() [3/3]

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator- (
    difference_type n ) const [inline]
```

Definition at line 257 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index, and Rcpp::ConstMatrixRow< RTYPE >↔::const\_iterator::row.

#### 6.92.4.10 operator--() [1/2]

```
template<int RTYPE>
const_iterator& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-- ( ) [inline]
```

Definition at line 246 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

#### 6.92.4.11 operator--() [2/2]

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-- (
    int ) [inline]
```

Definition at line 250 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

#### 6.92.4.12 operator-=( )

```
template<int RTYPE>
const_iterator& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-= (
    difference_type n ) [inline]
```

Definition at line 261 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

#### 6.92.4.13 operator->( )

```
template<int RTYPE>
const pointer Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator-> ( ) [inline]
```

Definition at line 266 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index, and Rcpp::ConstMatrixRow< RTYPE >↔::const\_iterator::row.

#### 6.92.4.14 operator<( )

```
template<int RTYPE>
bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator< (
    const const_iterator & other ) [inline]
```

Definition at line 272 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

#### 6.92.4.15 operator<=( )

```
template<int RTYPE>
bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator<= (
    const const_iterator & other ) [inline]
```

Definition at line 274 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

#### 6.92.4.16 operator==( )

```
template<int RTYPE>
bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator==(
    const const_iterator & other ) [inline]
```

Definition at line 270 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

#### 6.92.4.17 operator>( )

```
template<int RTYPE>
bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator>(
    const const_iterator & other ) [inline]
```

Definition at line 273 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

#### 6.92.4.18 operator>=( )

```
template<int RTYPE>
bool Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator>=(
    const const_iterator & other ) [inline]
```

Definition at line 275 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index.

#### 6.92.4.19 operator[]( )

```
template<int RTYPE>
const reference Rcpp::ConstMatrixRow< RTYPE >::const_iterator::operator[] (
    int i ) const [inline]
```

Definition at line 277 of file MatrixRow.h.

References Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::index, and Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::row.

## 6.92.5 Member Data Documentation

### 6.92.5.1 index

```
template<int RTYPE>
int Rcpp::ConstMatrixRow< RTYPE >::const_iterator::index [private]
```

Definition at line 287 of file MatrixRow.h.

Referenced by Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator!=(()), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator\*(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator+(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator++(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator+=(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator-(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator--(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator-=(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator->(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator<(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator<=(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator==(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator>(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator>=(), and Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator[]().

### 6.92.5.2 row

```
template<int RTYPE>
const ConstMatrixRow& Rcpp::ConstMatrixRow< RTYPE >::const_iterator::row [private]
```

Definition at line 286 of file MatrixRow.h.

Referenced by Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator\*(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator+(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator-(), Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator->(), and Rcpp::ConstMatrixRow< RTYPE >::const\_iterator::operator[]().

The documentation for this class was generated from the following file:

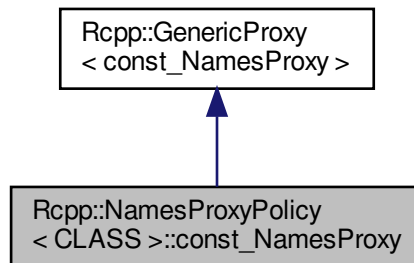
- inst/include/Rcpp/vector/[MatrixRow.h](#)



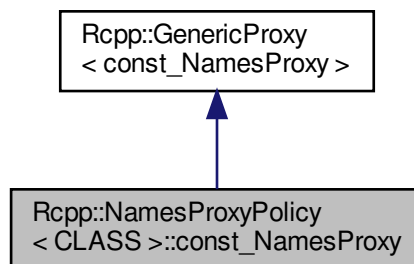
## 6.93 Rcpp::NamesProxyPolicy< CLASS >::const\_NamesProxy Class Reference

```
#include <NamesProxy.h>
```

Inheritance diagram for Rcpp::NamesProxyPolicy< CLASS >::const\_NamesProxy:



Collaboration diagram for Rcpp::NamesProxyPolicy< CLASS >::const\_NamesProxy:



### Public Member Functions

- [const\\_NamesProxy](#) (const CLASS &v)
- [template<typename T > operator T \(\) const](#)

## Private Member Functions

- SEXP [get](#) () const

## Private Attributes

- const CLASS & [parent](#)

### 6.93.1 Detailed Description

```
template<typename CLASS>  
class Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy
```

Definition at line 67 of file NamesProxy.h.

### 6.93.2 Constructor & Destructor Documentation

#### 6.93.2.1 const\_NamesProxy()

```
template<typename CLASS >  
Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::const_NamesProxy (  
    const CLASS & v ) [inline]
```

Definition at line 69 of file NamesProxy.h.

### 6.93.3 Member Function Documentation

#### 6.93.3.1 get()

```
template<typename CLASS >  
SEXP Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::get ( ) const [inline], [private]
```

Definition at line 76 of file NamesProxy.h.

References [Rcpp::NamesProxyPolicy< CLASS >::const\\_NamesProxy::parent](#), and [RCPP\\_GET\\_NAMES](#).

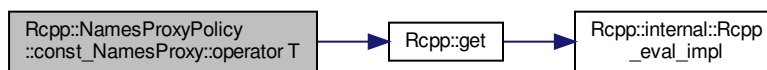
### 6.93.3.2 operator T()

```
template<typename CLASS >  
template<typename T >  
Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::operator T
```

Definition at line 76 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



## 6.93.4 Member Data Documentation

### 6.93.4.1 parent

```
template<typename CLASS >  
const CLASS& Rcpp::NamesProxyPolicy< CLASS >::const_NamesProxy::parent [private]
```

Definition at line 74 of file NamesProxy.h.

Referenced by Rcpp::NamesProxyPolicy< CLASS >::const\_NamesProxy::get().

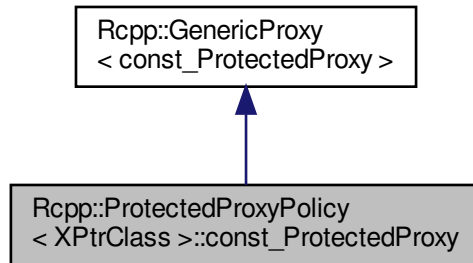
The documentation for this class was generated from the following files:

- [inst/include/Rcpp/proxy/NamesProxy.h](#)
- [inst/include/Rcpp/api/meat/proxy.h](#)

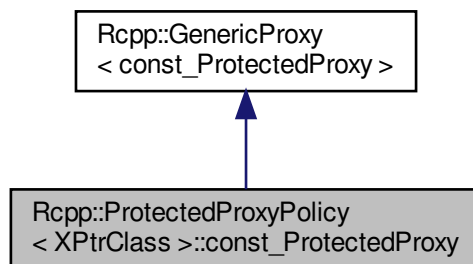
## 6.94 Rcpp::ProtectedProxyPolicy< XPtrClass >::const\_ProtectedProxy Class Reference

```
#include <ProtectedProxy.h>
```

Inheritance diagram for Rcpp::ProtectedProxyPolicy< XPtrClass >::const\_ProtectedProxy:



Collaboration diagram for Rcpp::ProtectedProxyPolicy< XPtrClass >::const\_ProtectedProxy:



### Public Member Functions

- [const\\_ProtectedProxy](#) (const XPtrClass &xp\_)
- `template<typename U >`  
[operator U](#) () const
- [operator SEXP](#) () const

## Private Member Functions

- SEXP [get](#) () const

## Private Attributes

- const XPtrClass & [xp](#)

### 6.94.1 Detailed Description

```
template<class XPtrClass>
class Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy
```

Definition at line 61 of file ProtectedProxy.h.

### 6.94.2 Constructor & Destructor Documentation

#### 6.94.2.1 const\_ProtectedProxy()

```
template<class XPtrClass >
Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::const_ProtectedProxy (
    const XPtrClass & xp_ ) [inline]
```

Definition at line 63 of file ProtectedProxy.h.

### 6.94.3 Member Function Documentation

#### 6.94.3.1 get()

```
template<class XPtrClass >
SEXP Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::get ( ) const [inline],
[private]
```

Definition at line 77 of file ProtectedProxy.h.

References [Rcpp::ProtectedProxyPolicy< XPtrClass >::const\\_ProtectedProxy::xp](#).

Referenced by [Rcpp::ProtectedProxyPolicy< XPtrClass >::const\\_ProtectedProxy::operator SEXP\(\)](#), and [Rcpp::ProtectedProxyPolicy< XPtrClass >::const\\_ProtectedProxy::operator U\(\)](#).

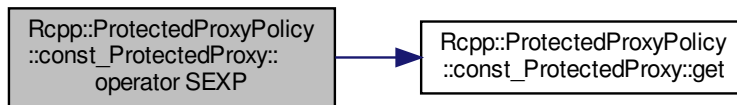
### 6.94.3.2 operator SEXP()

```
template<class XPtrClass >
Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::operator SEXP ( ) const [inline]
```

Definition at line 70 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::const\_ProtectedProxy::get().

Here is the call graph for this function:



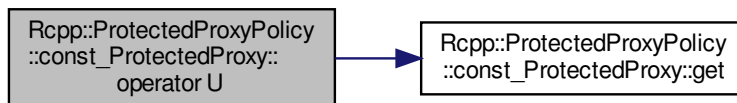
### 6.94.3.3 operator U()

```
template<class XPtrClass >
template<typename U >
Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::operator U ( ) const [inline]
```

Definition at line 66 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::const\_ProtectedProxy::get().

Here is the call graph for this function:



## 6.94.4 Member Data Documentation

## 6.94.4.1 xp

```
template<class XPtrClass >
const XPtrClass& Rcpp::ProtectedProxyPolicy< XPtrClass >::const_ProtectedProxy::xp [private]
```

Definition at line 75 of file ProtectedProxy.h.

Referenced by Rcpp::ProtectedProxyPolicy< XPtrClass >::const\_ProtectedProxy::get().

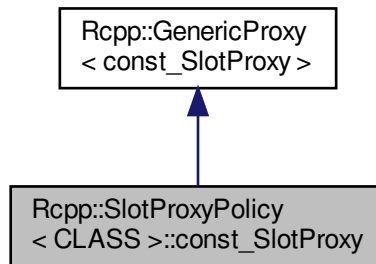
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/ProtectedProxy.h

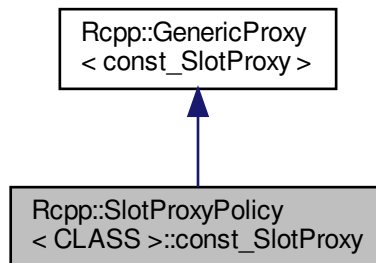
## 6.95 Rcpp::SlotProxyPolicy&lt; CLASS &gt;::const\_SlotProxy Class Reference

```
#include <SlotProxy.h>
```

Inheritance diagram for Rcpp::SlotProxyPolicy< CLASS >::const\_SlotProxy:



Collaboration diagram for Rcpp::SlotProxyPolicy< CLASS >::const\_SlotProxy:



## Public Member Functions

- [const\\_SlotProxy](#) (const CLASS &v, const std::string &name)
- `template<typename T >`  
[operator T](#) () const
- [operator SEXP](#) () const

## Private Member Functions

- [SEXP get](#) () const

## Private Attributes

- const CLASS & [parent](#)
- SEXP [slot\\_name](#)

### 6.95.1 Detailed Description

```
template<typename CLASS>  
class Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy
```

Definition at line 59 of file SlotProxy.h.

### 6.95.2 Constructor & Destructor Documentation

#### 6.95.2.1 const\_SlotProxy()

```
template<typename CLASS >  
Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::const_SlotProxy (  
    const CLASS & v,  
    const std::string & name ) [inline]
```

Definition at line 61 of file SlotProxy.h.

References [Rcpp::SlotProxyPolicy< CLASS >::const\\_SlotProxy::slot\\_name](#).

### 6.95.3 Member Function Documentation



### 6.95.3.1 get()

```
template<typename CLASS >
SEXP Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::get ( ) const [inline], [private]
```

Definition at line 78 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::const\_SlotProxy::parent, and Rcpp::SlotProxyPolicy< CLASS >::const\_SlotProxy::slot\_name.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::const\_SlotProxy::operator SEXP(), and Rcpp::SlotProxyPolicy< CLASS >::const\_SlotProxy::operator T().

### 6.95.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::operator SEXP ( ) const [inline]
```

Definition at line 70 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::const\_SlotProxy::get().

Here is the call graph for this function:



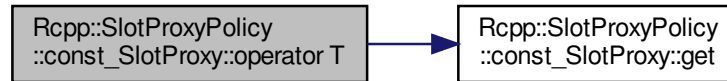
### 6.95.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::operator T ( ) const [inline]
```

Definition at line 67 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::const\_SlotProxy::get().

Here is the call graph for this function:



## 6.95.4 Member Data Documentation

### 6.95.4.1 parent

```

template<typename CLASS >
const CLASS& Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::parent [private]
  
```

Definition at line 75 of file SlotProxy.h.

Referenced by `Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::get()`.

### 6.95.4.2 slot\_name

```

template<typename CLASS >
SEXP Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::slot_name [private]
  
```

Definition at line 76 of file SlotProxy.h.

Referenced by `Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::const_SlotProxy()`, and `Rcpp::SlotProxyPolicy< CLASS >::const_SlotProxy::get()`.

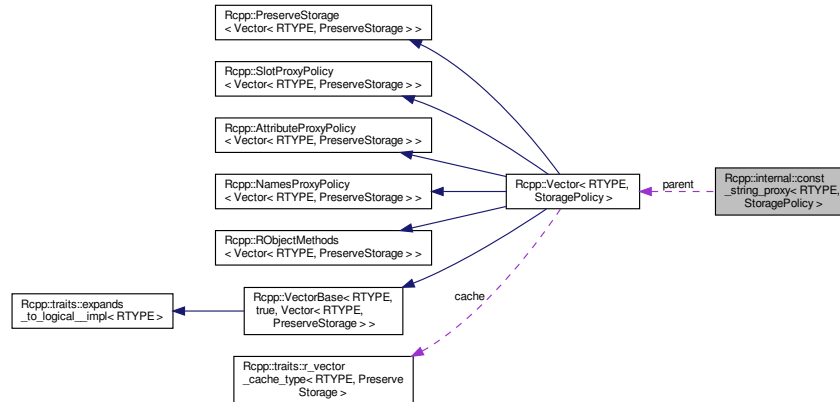
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/proxy/SlotProxy.h`

## 6.96 Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy > Class Template Reference

```
#include <const_string_proxy.h>
```

Collaboration diagram for Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >:



### Public Types

- typedef `::Rcpp::Vector< RTYPE, StoragePolicy >` `VECTOR`
- typedef `const char *` `iterator`
- typedef `const char &` `reference`

### Public Member Functions

- `const_string_proxy ()`
- `const_string_proxy (const VECTOR &v, R_xlen_t index_)`
- `const_string_proxy (SEXP x)`
- `const_string_proxy (const const_string_proxy &other)`
- `void import (const const_string_proxy &other)`
- `operator SEXP () const`
- `operator char * () const`
- `void move (R_xlen_t n)`
- `SEXP get () const`
- `iterator begin () const`
- `iterator end () const`
- `R_xlen_t size () const`
- `bool empty () const`
- `reference operator[] (R_xlen_t n)`
- `bool operator== (const char *other)`
- `bool operator!= (const char *other)`
- `bool operator== (const const_string_proxy &other)`
- `bool operator!= (const const_string_proxy &other)`
- `bool operator== (SEXP other) const`
- `bool operator!= (SEXP other) const`

## Public Attributes

- const [VECTOR](#) \* [parent](#)
- [R\\_xlen\\_t](#) [index](#)

## Static Private Attributes

- static std::string [buffer](#)

## Friends

- `template<int RT, template< class > class StoragePolicy_>  
std::ostream & operator<< (std::ostream &os, const const\_string\_proxy< RT, StoragePolicy_ > &proxy)`
- `template<int RT, template< class > class StoragePolicy_>  
std::string operator+ (const std::string &x, const const\_string\_proxy< RT, StoragePolicy_ > &proxy)`

### 6.96.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>
class Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >
```

Definition at line 29 of file `const_string_proxy.h`.

### 6.96.2 Member Typedef Documentation

#### 6.96.2.1 iterator

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const char* Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::iterator
```

Definition at line 33 of file `const_string_proxy.h`.

#### 6.96.2.2 reference

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const char& Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::reference
```

Definition at line 34 of file `const_string_proxy.h`.

### 6.96.2.3 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::VECTOR
```

Definition at line 32 of file const\_string\_proxy.h.

## 6.96.3 Constructor & Destructor Documentation

### 6.96.3.1 const\_string\_proxy() [1/4]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::const_string_proxy ( ) [inline]
```

Definition at line 36 of file const\_string\_proxy.h.

### 6.96.3.2 const\_string\_proxy() [2/4]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::const_string_proxy (
    const VECTOR & v,
    R_xlen_t index_ ) [inline]
```

Creates a proxy

#### Parameters

<i>v</i>	reference to the associated character vector
<i>index</i>	index

Definition at line 44 of file const\_string\_proxy.h.

### 6.96.3.3 const\_string\_proxy() [3/4]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::const_string_proxy (
    SEXP x ) [inline]
```

Definition at line 46 of file `const_string_proxy.h`.

References `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::parent`.

#### 6.96.3.4 `const_string_proxy()` [4/4]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::const_string_proxy (
    const const_string_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 51 of file `const_string_proxy.h`.

### 6.96.4 Member Function Documentation

#### 6.96.4.1 `begin()`

```
template<int RTYPE, template< class > class StoragePolicy>
iterator Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin ( ) const [inline]
```

Definition at line 94 of file `const_string_proxy.h`.

References `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::index`, and `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::parent`.

Referenced by `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::empty()`, `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::end()`, `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator!=(())`, `Rcpp::internal::operator<()`, `Rcpp::internal::operator<=()`, `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator==(())`, `Rcpp::internal::operator>()`, `Rcpp::internal::operator>=()`, `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator[]()`, and `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::size()`.

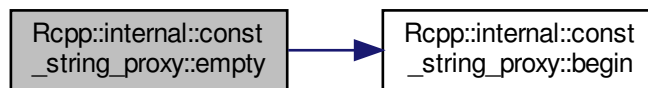
### 6.96.4.2 empty()

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::empty ( ) const [inline]
```

Definition at line 97 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



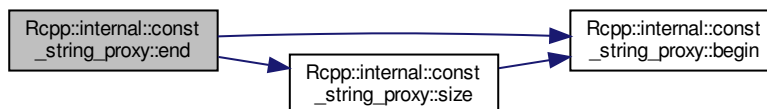
### 6.96.4.3 end()

```
template<int RTYPE, template< class > class StoragePolicy>
iterator Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::end ( ) const [inline]
```

Definition at line 95 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::begin(), and Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



#### 6.96.4.4 get()

```
template<int RTYPE, template< class > class StoragePolicy>
SEXPRcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::get ( ) const [inline]
```

Definition at line 90 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::operator char \*(), Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::operator SEXP(), Rcpp::String::operator!==( ), Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::operator!==( ), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::operator=( ), Rcpp::String::operator==( ), and Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::operator==( ).

#### 6.96.4.5 import()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::import (
    const const_string_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 54 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::parent.

#### 6.96.4.6 move()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::move (
    R_xlen_t n ) [inline]
```

Definition at line 88 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::index.



#### 6.96.4.7 operator char \*()

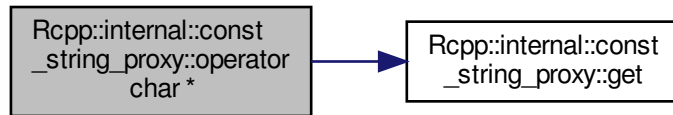
```
template<int RTYPE, template< class > class StoragePolicy>  
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator char * ( ) const [inline]
```

rhs use. Retrieves the current value of the element this proxy refers to and convert it to a C string

Definition at line 72 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



#### 6.96.4.8 operator SEXP()

```
template<int RTYPE, template< class > class StoragePolicy>  
Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const [inline]
```

rhs use. Retrieves the current value of the element this proxy refers to.

Definition at line 63 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



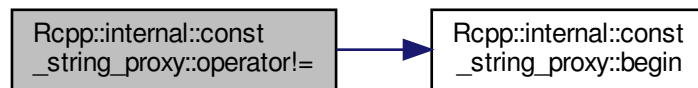
**6.96.4.9 operator"!=(()) [1/3]**

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator!= (
    const char * other ) [inline]
```

Definition at line 103 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:

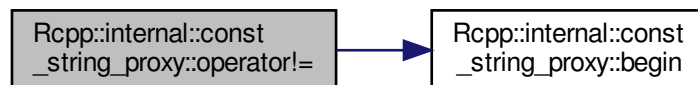
**6.96.4.10 operator"!=(()) [2/3]**

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator!= (
    const const_string_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 110 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



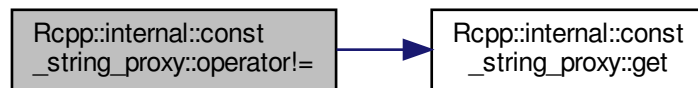
#### 6.96.4.11 operator"!==( ) [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator!= (
    SEXP other ) const [inline]
```

Definition at line 118 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



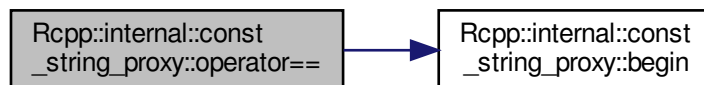
#### 6.96.4.12 operator==( ) [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator==(
    const char * other ) [inline]
```

Definition at line 100 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



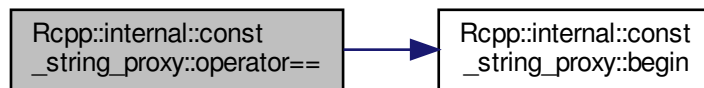
**6.96.4.13 operator==( ) [2/3]**

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator==(
    const const_string_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 107 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:

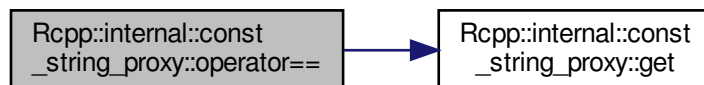
**6.96.4.14 operator==( ) [3/3]**

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator==(
    SEXP other ) const [inline]
```

Definition at line 114 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



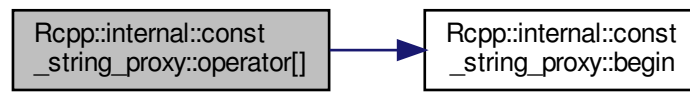
#### 6.96.4.15 operator[]()

```
template<int RTYPE, template< class > class StoragePolicy>
reference Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::operator[] (
    R_xlen_t n) [inline]
```

Definition at line 98 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



#### 6.96.4.16 size()

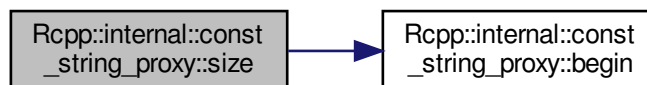
```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::size ( ) const [inline]
```

Definition at line 96 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::begin().

Referenced by Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



## 6.96.5 Friends And Related Function Documentation

### 6.96.5.1 operator+

```
template<int RTYPE, template< class > class StoragePolicy>
template<int RT, template< class > class StoragePolicy_>
std::string operator+ (
    const std::string & x,
    const const\_string\_proxy< RT, StoragePolicy_ > & proxy ) [friend]
```

### 6.96.5.2 operator<<

```
template<int RTYPE, template< class > class StoragePolicy>
template<int RT, template< class > class StoragePolicy_>
std::ostream& operator<< (
    std::ostream & os,
    const const\_string\_proxy< RT, StoragePolicy_ > & proxy ) [friend]
```

Prints the element this proxy refers to an output stream

## 6.96.6 Member Data Documentation

### 6.96.6.1 buffer

```
template<int RTYPE, template< class > class StoragePolicy>
std::string Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::buffer [static], [private]
```

Definition at line 123 of file `const_string_proxy.h`.

### 6.96.6.2 index

```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::index
```

Definition at line 87 of file `const_string_proxy.h`.

Referenced by `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::begin()`, `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::get()`, `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::import()`, and `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::move()`.

### 6.96.6.3 parent

```
template<int RTYPE, template< class > class StoragePolicy>
const VECTOR* Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::parent
```

Definition at line 86 of file const\_string\_proxy.h.

Referenced by Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::begin(), Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::const\_string\_proxy(), Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::import().

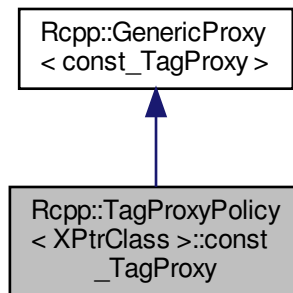
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00\_forward\_proxy.h
- inst/include/Rcpp/vector/const\_string\_proxy.h

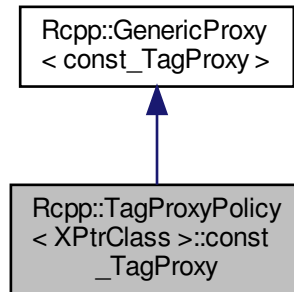
## 6.97 Rcpp::TagProxyPolicy< XPtrClass >::const\_TagProxy Class Reference

```
#include <TagProxy.h>
```

Inheritance diagram for Rcpp::TagProxyPolicy< XPtrClass >::const\_TagProxy:



Collaboration diagram for `Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy`:



## Public Member Functions

- [const\\_TagProxy](#) (XPtrClass &xp\_)
- `template<typename U >`  
[operator U](#) () const
- [operator SEXP](#) () const

## Private Member Functions

- `SEXP` [get](#) () const

## Private Attributes

- XPtrClass & [xp](#)

### 6.97.1 Detailed Description

```

template<typename XPtrClass>
class Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy
  
```

Definition at line 53 of file TagProxy.h.

### 6.97.2 Constructor & Destructor Documentation



### 6.97.2.1 const\_TagProxy()

```
template<typename XPtrClass >  
Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::const_TagProxy (  
    XPtrClass & xp_ ) [inline]
```

Definition at line 55 of file TagProxy.h.

## 6.97.3 Member Function Documentation

### 6.97.3.1 get()

```
template<typename XPtrClass >  
SEXPR Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::get ( ) const [inline], [private]
```

Definition at line 66 of file TagProxy.h.

References Rcpp::TagProxyPolicy< XPtrClass >::const\_TagProxy::xp.

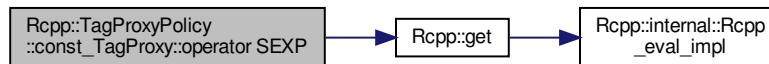
### 6.97.3.2 operator SEXP()

```
template<typename CLASS >  
Rcpp::TagProxyPolicy< CLASS >::const_TagProxy::operator SEXP
```

Definition at line 122 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



### 6.97.3.3 operator U()

```
template<typename XPtrClass >
template<typename U >
Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::operator U ( ) const
```

## 6.97.4 Member Data Documentation

### 6.97.4.1 xp

```
template<typename XPtrClass >
XPtrClass& Rcpp::TagProxyPolicy< XPtrClass >::const_TagProxy::xp [private]
```

Definition at line 64 of file TagProxy.h.

Referenced by Rcpp::TagProxyPolicy< XPtrClass >::const\_TagProxy::get().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/TagProxy.h
- inst/include/Rcpp/api/meat/proxy.h

## 6.98 Rcpp::ConstInputParameter< T > Class Template Reference

```
#include <InputParameter.h>
```

### Public Types

- typedef const T [const\\_nonref](#)

### Public Member Functions

- [ConstInputParameter](#) (SEXP x\_)
- [operator const\\_nonref](#) ()

### Private Attributes

- T [obj](#)

## 6.98.1 Detailed Description

```
template<typename T>  
class Rcpp::ConstInputParameter< T >
```

Definition at line 56 of file InputParameter.h.

## 6.98.2 Member Typedef Documentation

### 6.98.2.1 const\_nonref

```
template<typename T >  
typedef const T Rcpp::ConstInputParameter< T >::const_nonref
```

Definition at line 58 of file InputParameter.h.

## 6.98.3 Constructor & Destructor Documentation

### 6.98.3.1 ConstInputParameter()

```
template<typename T >  
Rcpp::ConstInputParameter< T >::ConstInputParameter (   
    SEXP x_ ) [inline]
```

Definition at line 59 of file InputParameter.h.

## 6.98.4 Member Function Documentation

### 6.98.4.1 operator const\_nonref()

```
template<typename T >  
Rcpp::ConstInputParameter< T >::operator const_nonref ( ) [inline]
```

Definition at line 61 of file InputParameter.h.

References Rcpp::ConstInputParameter< T >::obj.

## 6.98.5 Member Data Documentation

### 6.98.5.1 obj

```
template<typename T >
T Rcpp::ConstInputParameter< T >::obj [private]
```

Definition at line 64 of file InputParameter.h.

Referenced by Rcpp::ConstInputParameter< T >::operator const\_nonref().

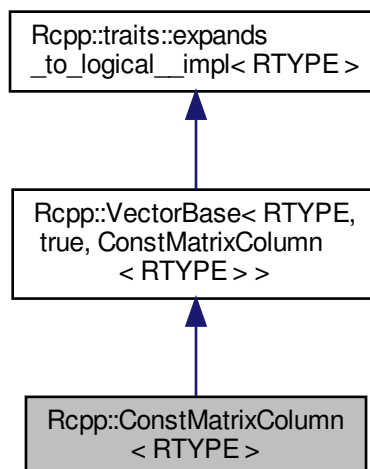
The documentation for this class was generated from the following file:

- inst/include/Rcpp/InputParameter.h

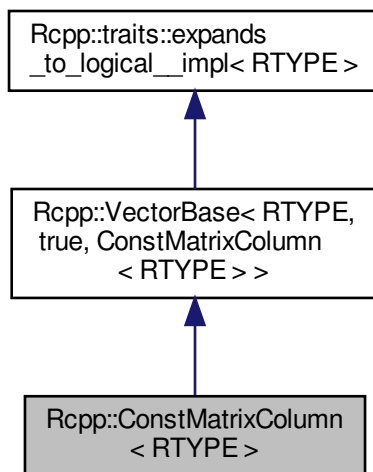
## 6.99 Rcpp::ConstMatrixColumn< RTYPE > Class Template Reference

```
#include <MatrixColumn.h>
```

Inheritance diagram for Rcpp::ConstMatrixColumn< RTYPE >:



Collaboration diagram for Rcpp::ConstMatrixColumn< RTYPE >:



## Public Types

- typedef `Matrix< RTYPE > MATRIX`
- typedef `MATRIX::const_Proxy const_Proxy`
- typedef `MATRIX::value_type value_type`
- typedef `MATRIX::const_iterator const_iterator`

## Public Member Functions

- `ConstMatrixColumn` (const `MATRIX` &parent, int i)
- `ConstMatrixColumn` (const `ConstMatrixColumn` &other)
- `const_Proxy operator[]` (int i) const
- `const_iterator begin` () const
- `const_iterator end` () const
- `const_iterator cbegin` () const
- `const_iterator cend` () const
- int `size` () const

## Private Attributes

- const int n
- `const_iterator const_start`

### 6.99.1 Detailed Description

```
template<int RTYPE>  
class Rcpp::ConstMatrixColumn< RTYPE >
```

Definition at line 123 of file MatrixColumn.h.

### 6.99.2 Member Typedef Documentation

#### 6.99.2.1 const\_iterator

```
template<int RTYPE>  
typedef MATRIX::const_iterator Rcpp::ConstMatrixColumn< RTYPE >::const_iterator
```

Definition at line 128 of file MatrixColumn.h.

#### 6.99.2.2 const\_Proxy

```
template<int RTYPE>  
typedef MATRIX::const_Proxy Rcpp::ConstMatrixColumn< RTYPE >::const_Proxy
```

Definition at line 126 of file MatrixColumn.h.

#### 6.99.2.3 MATRIX

```
template<int RTYPE>  
typedef Matrix<RTYPE> Rcpp::ConstMatrixColumn< RTYPE >::MATRIX
```

Definition at line 125 of file MatrixColumn.h.

#### 6.99.2.4 value\_type

```
template<int RTYPE>  
typedef MATRIX::value_type Rcpp::ConstMatrixColumn< RTYPE >::value_type
```

Definition at line 127 of file MatrixColumn.h.

### 6.99.3 Constructor & Destructor Documentation

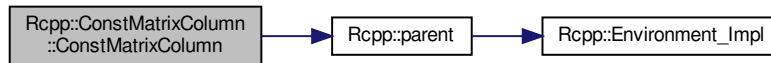
#### 6.99.3.1 ConstMatrixColumn() [1/2]

```
template<int RTYPE>
Rcpp::ConstMatrixColumn< RTYPE >::ConstMatrixColumn (
    const MATRIX & parent,
    int i ) [inline]
```

Definition at line 130 of file MatrixColumn.h.

References Rcpp::parent().

Here is the call graph for this function:



#### 6.99.3.2 ConstMatrixColumn() [2/2]

```
template<int RTYPE>
Rcpp::ConstMatrixColumn< RTYPE >::ConstMatrixColumn (
    const ConstMatrixColumn< RTYPE > & other ) [inline]
```

Definition at line 141 of file MatrixColumn.h.

### 6.99.4 Member Function Documentation

#### 6.99.4.1 begin()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixColumn< RTYPE >::begin ( ) const [inline]
```

Definition at line 149 of file MatrixColumn.h.

References Rcpp::ConstMatrixColumn< RTYPE >::const\_start.

#### 6.99.4.2 cbegin()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixColumn< RTYPE >::cbegin ( ) const [inline]
```

Definition at line 157 of file MatrixColumn.h.

References Rcpp::ConstMatrixColumn< RTYPE >::const\_start.

#### 6.99.4.3 cend()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixColumn< RTYPE >::cend ( ) const [inline]
```

Definition at line 161 of file MatrixColumn.h.

References Rcpp::ConstMatrixColumn< RTYPE >::const\_start, and Rcpp::ConstMatrixColumn< RTYPE >::n.

#### 6.99.4.4 end()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixColumn< RTYPE >::end ( ) const [inline]
```

Definition at line 153 of file MatrixColumn.h.

References Rcpp::ConstMatrixColumn< RTYPE >::const\_start, and Rcpp::ConstMatrixColumn< RTYPE >::n.

#### 6.99.4.5 operator[]()

```
template<int RTYPE>
const_Proxy Rcpp::ConstMatrixColumn< RTYPE >::operator[] (
    int i ) const [inline]
```

Definition at line 145 of file MatrixColumn.h.

References Rcpp::ConstMatrixColumn< RTYPE >::const\_start.



### 6.99.4.6 size()

```
template<int RTYPE>
int Rcpp::ConstMatrixColumn< RTYPE >::size ( ) const [inline]
```

Definition at line 165 of file MatrixColumn.h.

References Rcpp::ConstMatrixColumn< RTYPE >::n.

## 6.99.5 Member Data Documentation

### 6.99.5.1 const\_start

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixColumn< RTYPE >::const_start [private]
```

Definition at line 171 of file MatrixColumn.h.

Referenced by Rcpp::ConstMatrixColumn< RTYPE >::begin(), Rcpp::ConstMatrixColumn< RTYPE >::cbegin(), Rcpp::ConstMatrixColumn< RTYPE >::cend(), Rcpp::ConstMatrixColumn< RTYPE >::end(), and Rcpp::ConstMatrixColumn< RTYPE >::operator[]().

### 6.99.5.2 n

```
template<int RTYPE>
const int Rcpp::ConstMatrixColumn< RTYPE >::n [private]
```

Definition at line 170 of file MatrixColumn.h.

Referenced by Rcpp::ConstMatrixColumn< RTYPE >::cend(), Rcpp::ConstMatrixColumn< RTYPE >::end(), and Rcpp::ConstMatrixColumn< RTYPE >::size().

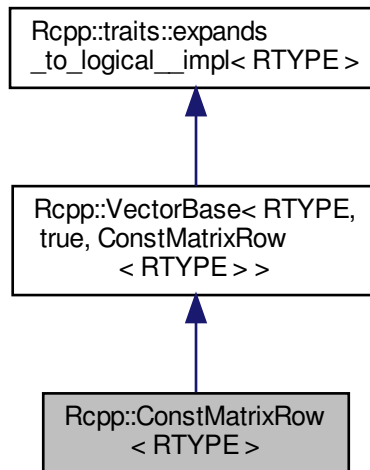
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixColumn.h

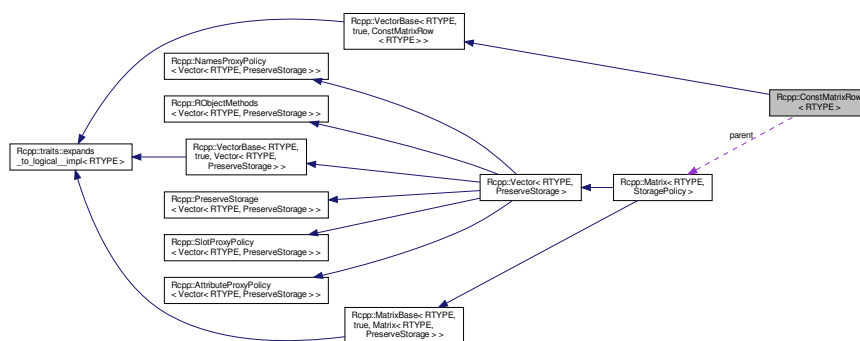
## 6.100 Rcpp::ConstMatrixRow< RTYPE > Class Template Reference

```
#include <MatrixRow.h>
```

Inheritance diagram for Rcpp::ConstMatrixRow< RTYPE >:



Collaboration diagram for Rcpp::ConstMatrixRow< RTYPE >:



### Classes

- class [const\\_iterator](#)

## Public Types

- typedef [Matrix< RTYPE > MATRIX](#)
- typedef [MATRIX::const\\_Proxy const\\_reference](#)
- typedef [MATRIX::value\\_type value\\_type](#)
- typedef [const\\_iterator iterator](#)

## Public Member Functions

- [ConstMatrixRow](#) (const [MATRIX](#) &object, int i)
- [ConstMatrixRow](#) (const [ConstMatrixRow](#) &other)
- [const\\_reference operator\[\]](#) (int i) const
- [const\\_iterator begin](#) () const
- [const\\_iterator end](#) () const
- int [size](#) () const

## Private Member Functions

- int [get\\_parent\\_index](#) (int i) const

## Private Attributes

- const [MATRIX](#) & [parent](#)
- [MATRIX::const\\_iterator](#) [start](#)
- int [parent\\_nrow](#)
- int [row](#)

### 6.100.1 Detailed Description

```
template<int RTYPE>
class Rcpp::ConstMatrixRow< RTYPE >
```

Definition at line 216 of file [MatrixRow.h](#).

### 6.100.2 Member Typedef Documentation

#### 6.100.2.1 const\_reference

```
template<int RTYPE>
typedef MATRIX::const\_Proxy Rcpp::ConstMatrixRow< RTYPE >::const_reference
```

Definition at line 219 of file [MatrixRow.h](#).

### 6.100.2.2 iterator

```
template<int RTYPE>
typedef const\_iterator Rcpp::ConstMatrixRow< RTYPE >::iterator
```

Definition at line 290 of file [MatrixRow.h](#).

### 6.100.2.3 MATRIX

```
template<int RTYPE>
typedef Matrix<RTYPE> Rcpp::ConstMatrixRow< RTYPE >::MATRIX
```

Definition at line 218 of file [MatrixRow.h](#).

### 6.100.2.4 value\_type

```
template<int RTYPE>
typedef MATRIX::value\_type Rcpp::ConstMatrixRow< RTYPE >::value_type
```

Definition at line 220 of file [MatrixRow.h](#).

## 6.100.3 Constructor & Destructor Documentation

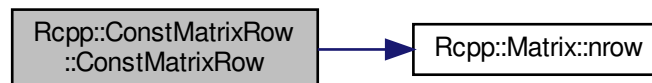
### 6.100.3.1 ConstMatrixRow() [1/2]

```
template<int RTYPE>
Rcpp::ConstMatrixRow< RTYPE >::ConstMatrixRow (
    const MATRIX & object,
    int i ) [inline]
```

Definition at line 292 of file [MatrixRow.h](#).

References [Rcpp::Matrix](#)< RTYPE, [StoragePolicy](#) >::nrow(), and [Rcpp::ConstMatrixRow](#)< RTYPE >::parent.

Here is the call graph for this function:



### 6.100.3.2 ConstMatrixRow() [2/2]

```
template<int RTYPE>
Rcpp::ConstMatrixRow< RTYPE >::ConstMatrixRow (
    const ConstMatrixRow< RTYPE > & other ) [inline]
```

Definition at line 305 of file MatrixRow.h.

## 6.100.4 Member Function Documentation

### 6.100.4.1 begin()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixRow< RTYPE >::begin ( ) const [inline]
```

Definition at line 316 of file MatrixRow.h.

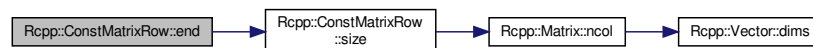
### 6.100.4.2 end()

```
template<int RTYPE>
const_iterator Rcpp::ConstMatrixRow< RTYPE >::end ( ) const [inline]
```

Definition at line 320 of file MatrixRow.h.

References `Rcpp::ConstMatrixRow< RTYPE >::size()`.

Here is the call graph for this function:



### 6.100.4.3 `get_parent_index()`

```
template<int RTYPE>
int Rcpp::ConstMatrixRow< RTYPE >::get_parent_index (
    int i ) const [inline], [private]
```

Definition at line 334 of file MatrixRow.h.

References `Rcpp::ConstMatrixRow< RTYPE >::parent_nrow`, and `RCPP_DEBUG_4`.

### 6.100.4.4 `operator[]()`

```
template<int RTYPE>
const_reference Rcpp::ConstMatrixRow< RTYPE >::operator[] (
    int i ) const [inline]
```

Definition at line 312 of file MatrixRow.h.

References `Rcpp::ConstMatrixRow< RTYPE >::parent`, `Rcpp::ConstMatrixRow< RTYPE >::parent_nrow`, and `Rcpp::ConstMatrixRow< RTYPE >::row`.

### 6.100.4.5 `size()`

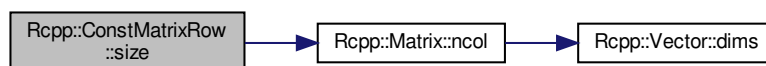
```
template<int RTYPE>
int Rcpp::ConstMatrixRow< RTYPE >::size ( ) const [inline]
```

Definition at line 324 of file MatrixRow.h.

References `Rcpp::Matrix< RTYPE, StoragePolicy >::ncol()`, and `Rcpp::ConstMatrixRow< RTYPE >::parent`.

Referenced by `Rcpp::ConstMatrixRow< RTYPE >::end()`.

Here is the call graph for this function:



## 6.100.5 Member Data Documentation

### 6.100.5.1 parent

```
template<int RTYPE>
const MATRIX& Rcpp::ConstMatrixRow< RTYPE >::parent [private]
```

Definition at line 329 of file MatrixRow.h.

Referenced by `Rcpp::ConstMatrixRow< RTYPE >::ConstMatrixRow()`, `Rcpp::ConstMatrixRow< RTYPE >::operator[]()`, and `Rcpp::ConstMatrixRow< RTYPE >::size()`.

### 6.100.5.2 parent\_nrow

```
template<int RTYPE>
int Rcpp::ConstMatrixRow< RTYPE >::parent_nrow [private]
```

Definition at line 331 of file MatrixRow.h.

Referenced by `Rcpp::ConstMatrixRow< RTYPE >::get_parent_index()`, and `Rcpp::ConstMatrixRow< RTYPE >::operator[]()`.

### 6.100.5.3 row

```
template<int RTYPE>
int Rcpp::ConstMatrixRow< RTYPE >::row [private]
```

Definition at line 332 of file MatrixRow.h.

Referenced by `Rcpp::ConstMatrixRow< RTYPE >::operator[]()`.

### 6.100.5.4 start

```
template<int RTYPE>
MATRIX::const_iterator Rcpp::ConstMatrixRow< RTYPE >::start [private]
```

Definition at line 330 of file MatrixRow.h.

The documentation for this class was generated from the following file:

- `inst/include/Rcpp/vector/MatrixRow.h`

## 6.101 Rcpp::ConstReferenceInputParameter< T > Class Template Reference

```
#include <InputParameter.h>
```

### Public Types

- typedef const T & [const\\_reference](#)

### Public Member Functions

- [ConstReferenceInputParameter](#) (SEXP x\_)
- [operator const\\_reference](#) ()

### Private Attributes

- T [obj](#)

#### 6.101.1 Detailed Description

```
template<typename T>  
class Rcpp::ConstReferenceInputParameter< T >
```

Definition at line 69 of file InputParameter.h.

#### 6.101.2 Member Typedef Documentation

##### 6.101.2.1 const\_reference

```
template<typename T >  
typedef const T& Rcpp::ConstReferenceInputParameter< T >::const\_reference
```

Definition at line 71 of file InputParameter.h.

#### 6.101.3 Constructor & Destructor Documentation



### 6.101.3.1 ConstReferenceInputParameter()

```
template<typename T >
Rcpp::ConstReferenceInputParameter< T >::ConstReferenceInputParameter (
    SEXP x_ ) [inline]
```

Definition at line 72 of file InputParameter.h.

## 6.101.4 Member Function Documentation

### 6.101.4.1 operator const\_reference()

```
template<typename T >
Rcpp::ConstReferenceInputParameter< T >::operator const_reference ( ) [inline]
```

Definition at line 74 of file InputParameter.h.

References Rcpp::ConstReferenceInputParameter< T >::obj.

## 6.101.5 Member Data Documentation

### 6.101.5.1 obj

```
template<typename T >
T Rcpp::ConstReferenceInputParameter< T >::obj [private]
```

Definition at line 77 of file InputParameter.h.

Referenced by Rcpp::ConstReferenceInputParameter< T >::operator const\_reference().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/InputParameter.h

## 6.102 Rcpp::traits::container\_exporter< Container, double > Struct Template Reference

```
#include <export.h>
```

## Public Types

- typedef [ContainerExporter](#)< Container, double > [type](#)

### 6.102.1 Detailed Description

```
template<template< class, class > class Container>
struct Rcpp::traits::container_exporter< Container, double >
```

Definition at line 65 of file export.h.

### 6.102.2 Member Typedef Documentation

#### 6.102.2.1 type

```
template<template< class, class > class Container>
typedef ContainerExporter< Container, double > Rcpp::traits::container_exporter< Container, double
>::type
```

Definition at line 66 of file export.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/meat/[export.h](#)

## 6.103 Rcpp::traits::container\_exporter< Container, int > Struct Template Reference

```
#include <export.h>
```

## Public Types

- typedef [ContainerExporter](#)< Container, int > [type](#)

### 6.103.1 Detailed Description

```
template<template< class, class > class Container>
struct Rcpp::traits::container_exporter< Container, int >
```

Definition at line 62 of file export.h.

## 6.103.2 Member Typedef Documentation

### 6.103.2.1 type

```
template<template< class, class > class Container>
typedef ContainerExporter< Container, int > Rcpp::traits::container_exporter< Container, int >↔
::type
```

Definition at line 63 of file export.h.

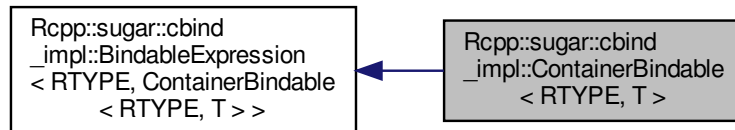
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/meat/export.h

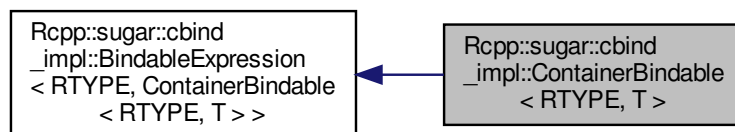
## 6.104 Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T > Class Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >:



Collaboration diagram for Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >:



## Public Types

- typedef [cbind\\_storage\\_type](#)< RTYPE >::type [stored\\_type](#)

## Public Member Functions

- [ContainerBindable](#) (const [Rcpp::Matrix](#)< RTYPE > &vec\_)
- [ContainerBindable](#) (const [Rcpp::Vector](#)< RTYPE > &vec\_)
- template<typename S >  
[ContainerBindable](#) (const [BindableExpression](#)< RTYPE, S > &e)
- [R\\_xlen\\_t size](#) () const
- [R\\_xlen\\_t nrow](#) () const
- [R\\_xlen\\_t ncol](#) () const
- [stored\\_type operator\[\]](#) (R\_xlen\_t i) const
- [stored\\_type operator\(\)](#) (R\_xlen\_t i, R\_xlen\_t j) const

## Private Attributes

- T [vec](#)
- [R\\_xlen\\_t len](#)
- [R\\_xlen\\_t nr](#)
- [R\\_xlen\\_t nc](#)

### 6.104.1 Detailed Description

```
template<int RTYPE, typename T >
class Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >
```

Definition at line 82 of file cbind.h.

### 6.104.2 Member Typedef Documentation

#### 6.104.2.1 stored\_type

```
template<int RTYPE, typename T >
typedef cbind\_storage\_type<RTYPE>::type Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >↔
::stored\_type
```

Definition at line 85 of file cbind.h.

### 6.104.3 Constructor & Destructor Documentation

#### 6.104.3.1 ContainerBindable() [1/3]

```
template<int RTYPE, typename T >
Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::ContainerBindable (
    const Rcpp::Matrix< RTYPE > & vec_ ) [inline]
```

Definition at line 92 of file cbind.h.

#### 6.104.3.2 ContainerBindable() [2/3]

```
template<int RTYPE, typename T >
Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::ContainerBindable (
    const Rcpp::Vector< RTYPE > & vec_ ) [inline]
```

Definition at line 97 of file cbind.h.

#### 6.104.3.3 ContainerBindable() [3/3]

```
template<int RTYPE, typename T >
template<typename S >
Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::ContainerBindable (
    const BindableExpression< RTYPE, S > & e ) [inline]
```

Definition at line 103 of file cbind.h.

References Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::len, and Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::vec.

### 6.104.4 Member Function Documentation

#### 6.104.4.1 ncol()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::ncol ( ) const [inline]
```

Definition at line 116 of file cbind.h.

References Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::nc.

#### 6.104.4.2 nrow()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::nrow ( ) const [inline]
```

Definition at line 114 of file cbind.h.

References Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::nr.

#### 6.104.4.3 operator()()

```
template<int RTYPE, typename T >
stored_type Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::operator() (
    R_xlen_t i,
    R_xlen_t j ) const [inline]
```

Definition at line 122 of file cbind.h.

References Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::nr, and Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::vec.

#### 6.104.4.4 operator[]()

```
template<int RTYPE, typename T >
stored_type Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 118 of file cbind.h.

References Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::vec.

#### 6.104.4.5 size()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::size ( ) const [inline]
```

Definition at line 112 of file cbind.h.

References Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::len.

## 6.104.5 Member Data Documentation

### 6.104.5.1 len

```
template<int RTYPE, typename T >  
R_xlen_t Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::len [private]
```

Definition at line 89 of file cbind.h.

Referenced by Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::ContainerBindable(), and Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::size().

### 6.104.5.2 nc

```
template<int RTYPE, typename T >  
R_xlen_t Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::nc [private]
```

Definition at line 89 of file cbind.h.

Referenced by Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::ncol().

### 6.104.5.3 nr

```
template<int RTYPE, typename T >  
R_xlen_t Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::nr [private]
```

Definition at line 89 of file cbind.h.

Referenced by Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::nrow(), and Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::operator()().

### 6.104.5.4 vec

```
template<int RTYPE, typename T >  
T Rcpp::sugar::cbind_impl::ContainerBindable< RTYPE, T >::vec [private]
```

Definition at line 88 of file cbind.h.

Referenced by Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::ContainerBindable(), Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::operator()(), and Rcpp::sugar::cbind\_impl::ContainerBindable< RTYPE, T >::operator[]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[cbind.h](#)

## 6.105 Rcpp::traits::ContainerExporter< ContainerTemplate, T > Class Template Reference

```
#include <export.h>
```

### Public Types

- typedef ContainerTemplate< T, std::allocator< T > > [Container](#)

### Public Member Functions

- [ContainerExporter](#) (SEXP x)
- [~ContainerExporter](#) ()
- [Container](#) get ()

### Static Public Attributes

- static const int [RTYPE](#) = [Rcpp::traits::r\\_sexptype\\_traits<T>::rtype](#)

### Private Attributes

- SEXP [object](#)

### 6.105.1 Detailed Description

```
template<template< class, class > class ContainerTemplate, typename T >
class Rcpp::traits::ContainerExporter< ContainerTemplate, T >
```

Definition at line 41 of file export.h.

### 6.105.2 Member Typedef Documentation

#### 6.105.2.1 Container

```
template<template< class, class > class ContainerTemplate, typename T >
typedef ContainerTemplate<T, std::allocator<T> > Rcpp::traits::ContainerExporter< Container←
Template, T >::Container
```

Definition at line 43 of file export.h.



## 6.105.3 Constructor & Destructor Documentation

### 6.105.3.1 ContainerExporter()

```
template<template< class, class > class ContainerTemplate, typename T >  
Rcpp::traits::ContainerExporter< ContainerTemplate, T >::ContainerExporter (  
    SEXP x ) [inline]
```

Definition at line 46 of file export.h.

### 6.105.3.2 ~ContainerExporter()

```
template<template< class, class > class ContainerTemplate, typename T >  
Rcpp::traits::ContainerExporter< ContainerTemplate, T >::~~ContainerExporter ( ) [inline]
```

Definition at line 47 of file export.h.

## 6.105.4 Member Function Documentation

### 6.105.4.1 get()

```
template<template< class, class > class ContainerTemplate, typename T >  
Container Rcpp::traits::ContainerExporter< ContainerTemplate, T >::get ( ) [inline]
```

Definition at line 49 of file export.h.

References Rcpp::traits::ContainerExporter< ContainerTemplate, T >::RTYPE.

## 6.105.5 Member Data Documentation

### 6.105.5.1 object

```
template<template< class, class > class ContainerTemplate, typename T >  
SEXP Rcpp::traits::ContainerExporter< ContainerTemplate, T >::object [private]
```

Definition at line 60 of file export.h.

### 6.105.5.2 RTYPE

```
template<template< class, class > class ContainerTemplate, typename T >
const int Rcpp::traits::ContainerExporter< ContainerTemplate, T >::RTYPE = Rcpp::traits::r_sexptype_traits<T>←
::rtype [static]
```

Definition at line 44 of file export.h.

Referenced by Rcpp::traits::ContainerExporter< ContainerTemplate, T >::get().

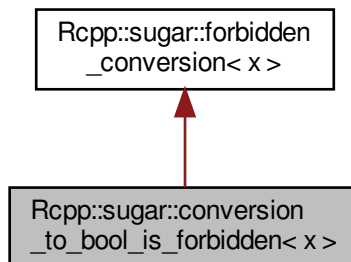
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/api/meat/export.h](#)

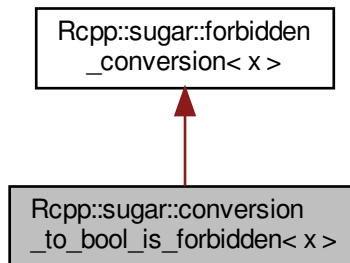
## 6.106 Rcpp::sugar::conversion\_to\_bool\_is\_forbidden< x > Class Template Reference

```
#include <SingleLogicalResult.h>
```

Inheritance diagram for Rcpp::sugar::conversion\_to\_bool\_is\_forbidden< x >:



Collaboration diagram for Rcpp::sugar::conversion\_to\_bool\_is\_forbidden< x >:



## Public Member Functions

- void [touch](#) ()

### 6.106.1 Detailed Description

```
template<bool x>
class Rcpp::sugar::conversion_to_bool_is_forbidden< x >
```

Definition at line 35 of file SingleLogicalResult.h.

### 6.106.2 Member Function Documentation

#### 6.106.2.1 touch()

```
template<bool x>
void Rcpp::sugar::conversion_to_bool_is_forbidden< x >::touch ( ) [inline]
```

Definition at line 38 of file SingleLogicalResult.h.

Referenced by Rcpp::sugar::SingleLogicalResult< NA, T >::operator bool().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/[SingleLogicalResult.h](#)

## 6.107 `tinyformat::detail::convertToInt< T, convertible >` Struct Template Reference

```
#include <tinyformat.h>
```

### Static Public Member Functions

- static int [invoke](#) (const T &)

#### 6.107.1 Detailed Description

```
template<typename T, bool convertible = is_convertible<T,int>::value>
struct tinyformat::detail::convertToInt< T, convertible >
```

Definition at line 253 of file `tinyformat.h`.

#### 6.107.2 Member Function Documentation

##### 6.107.2.1 `invoke()`

```
template<typename T , bool convertible = is_convertible<T,int>::value>
static int tinyformat::detail::convertToInt< T, convertible >::invoke (
    const T & ) [inline], [static]
```

Definition at line 255 of file `tinyformat.h`.

References `TINYFORMAT_ERROR`.

Referenced by `tinyformat::detail::FormatArg::toIntImpl()`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/utils/tinyformat/tinyformat.h](#)

## 6.108 `tinyformat::detail::convertToInt< T, true >` Struct Template Reference

```
#include <tinyformat.h>
```

## Static Public Member Functions

- static int [invoke](#) (const T &value)

### 6.108.1 Detailed Description

```
template<typename T>
struct tinyformat::detail::convertToInt< T, true >
```

Definition at line 264 of file tinyformat.h.

### 6.108.2 Member Function Documentation

#### 6.108.2.1 invoke()

```
template<typename T >
static int tinyformat::detail::convertToInt< T, true >::invoke (
    const T & value ) [inline], [static]
```

Definition at line 266 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/utils/tinyformat/tinyformat.h](#)

## 6.109 Rcpp::sugar::CountInserter< HASH, STORAGE > Class Template Reference

```
#include <table.h>
```

### Public Member Functions

- [CountInserter](#) (HASH &hash\_)
- void [operator\(\)](#) (STORAGE value)

### Private Attributes

- HASH & [hash](#)

### 6.109.1 Detailed Description

```
template<typename HASH, typename STORAGE>  
class Rcpp::sugar::CountInserter< HASH, STORAGE >
```

Definition at line 29 of file table.h.

### 6.109.2 Constructor & Destructor Documentation

#### 6.109.2.1 CountInserter()

```
template<typename HASH , typename STORAGE >  
Rcpp::sugar::CountInserter< HASH, STORAGE >::CountInserter (  
    HASH & hash_ ) [inline]
```

Definition at line 31 of file table.h.

### 6.109.3 Member Function Documentation

#### 6.109.3.1 operator>()

```
template<typename HASH , typename STORAGE >  
void Rcpp::sugar::CountInserter< HASH, STORAGE >::operator() (  
    STORAGE value ) [inline]
```

Definition at line 33 of file table.h.

References Rcpp::sugar::CountInserter< HASH, STORAGE >::hash.

### 6.109.4 Member Data Documentation

### 6.109.4.1 hash

```
template<typename HASH , typename STORAGE >  
HASH& Rcpp::sugar::CountInserter< HASH, STORAGE >::hash [private]
```

Definition at line 38 of file table.h.

Referenced by Rcpp::sugar::CountInserter< HASH, STORAGE >::operator()().

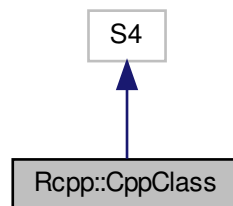
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[table.h](#)

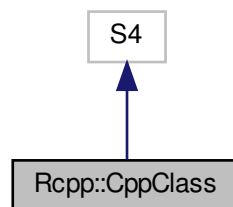
## 6.110 Rcpp::CppClass Class Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::CppClass:



Collaboration diagram for Rcpp::CppClass:



## Public Types

- typedef [XPtr](#)< [class\\_Base](#) > [XP\\_Class](#)
- typedef [Rcpp::XPtr](#)< [Rcpp::Module](#) > [XP](#)

## Public Member Functions

- [CppClass](#) (SEXP x)
- [CppClass](#) ([Module](#) \*p, [class\\_Base](#) \*cl, std::string &buffer)

## Private Types

- typedef [S4](#) [Base](#)

### 6.110.1 Detailed Description

Definition at line 382 of file [Module.h](#).

### 6.110.2 Member Typedef Documentation

#### 6.110.2.1 Base

```
typedef S4 Rcpp::CppClass::Base [private]
```

Definition at line 383 of file [Module.h](#).

#### 6.110.2.2 XP

```
typedef Rcpp::XPtr<Rcpp::Module> Rcpp::CppClass::XP
```

Definition at line 386 of file [Module.h](#).

#### 6.110.2.3 XP\_Class

```
typedef XPtr<class\_Base> Rcpp::CppClass::XP\_Class
```

Definition at line 385 of file [Module.h](#).



### 6.110.3 Constructor & Destructor Documentation

#### 6.110.3.1 CppClass() [1/2]

```
Rcpp::CppClass::CppClass (
    SEXP x ) [inline]
```

Definition at line 387 of file Module.h.

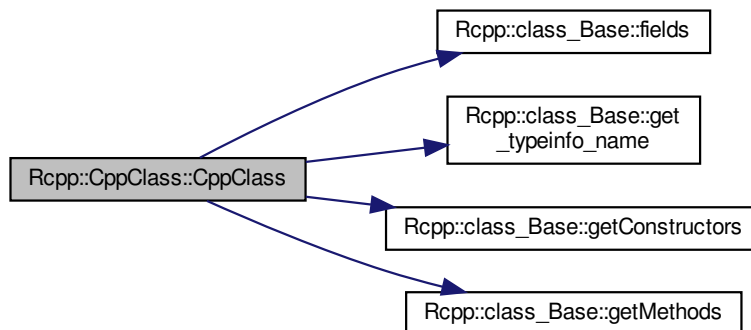
#### 6.110.3.2 CppClass() [2/2]

```
Rcpp::CppClass::CppClass (
    Module * p,
    class_Base * cl,
    std::string & buffer ) [inline]
```

Definition at line 389 of file Module.h.

References `Rcpp::class_Base::docstring`, `Rcpp::class_Base::enums`, `Rcpp::class_Base::fields()`, `Rcpp::class_Base::get_typeinfo_name()`, `Rcpp::class_Base::getConstructors()`, `Rcpp::class_Base::getMethods()`, `Rcpp::class_Base::name`, and `Rcpp::class_Base::parents`.

Here is the call graph for this function:

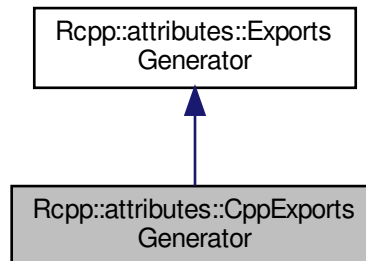


The documentation for this class was generated from the following file:

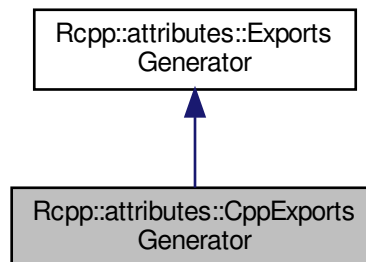
- [inst/include/Rcpp/Module.h](#)

## 6.111 Rcpp::attributes::CppExportsGenerator Class Reference

Inheritance diagram for Rcpp::attributes::CppExportsGenerator:



Collaboration diagram for Rcpp::attributes::CppExportsGenerator:



### Public Member Functions

- [CppExportsGenerator](#) (const std::string &packageDir, const std::string &package, const std::string &fileSep)
- virtual void [writeBegin](#) ()
- virtual void [writeEnd](#) (bool hasPackageInit)
- virtual bool [commit](#) (const std::vector< std::string > &includes)

### Private Member Functions

- virtual void [doWriteFunctions](#) (const [SourceFileAttributes](#) &attributes, bool verbose)
- std::string [registerCCallable](#) (size\_t indent, const std::string &exportedName, const std::string &name) const

## Private Attributes

- `std::vector< Attribute > initFunctions_`
- `std::vector< Attribute > cppExports_`
- `std::vector< Attribute > nativeRoutines_`
- `std::vector< std::string > modules_`

## Additional Inherited Members

### 6.111.1 Detailed Description

Definition at line 696 of file `attributes.cpp`.

### 6.111.2 Constructor & Destructor Documentation

#### 6.111.2.1 CppExportsGenerator()

```
Rcpp::attributes::CppExportsGenerator::CppExportsGenerator (
    const std::string & packageDir,
    const std::string & package,
    const std::string & fileSep ) [explicit]
```

Definition at line 1956 of file `attributes.cpp`.

### 6.111.3 Member Function Documentation

#### 6.111.3.1 commit()

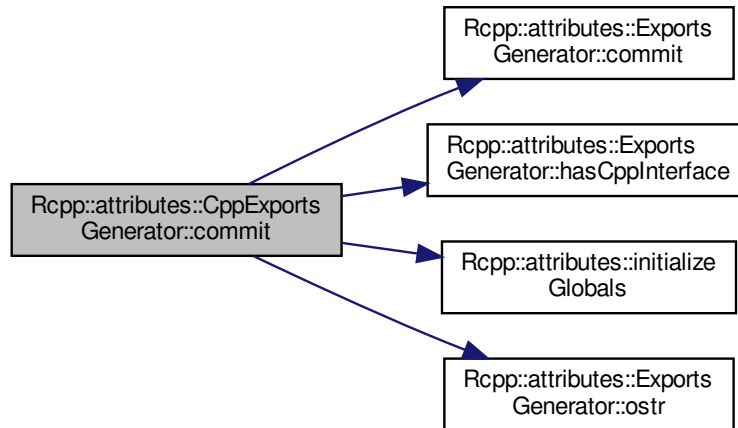
```
bool Rcpp::attributes::CppExportsGenerator::commit (
    const std::vector< std::string > & includes ) [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 2163 of file `attributes.cpp`.

References [Rcpp::attributes::ExportsGenerator::commit\(\)](#), [Rcpp::attributes::ExportsGenerator::hasCppInterface\(\)](#), [Rcpp::attributes::initializeGlobals\(\)](#), and [Rcpp::attributes::ExportsGenerator::ostr\(\)](#).

Here is the call graph for this function:



### 6.111.3.2 doWriteFunctions()

```

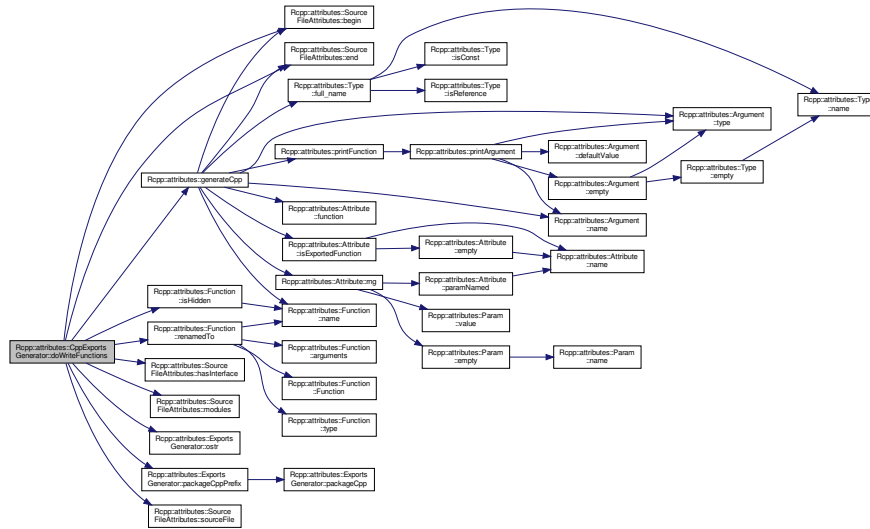
void Rcpp::attributes::CppExportsGenerator::doWriteFunctions (
    const SourceFileAttributes & attributes,
    bool verbose ) [private], [virtual]
  
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 1966 of file `attributes.cpp`.

References [Rcpp::attributes::SourceFileAttributes::begin\(\)](#), `cppExports_`, [Rcpp::attributes::SourceFileAttributes::end\(\)](#), [Rcpp::attributes::generateCpp\(\)](#), [Rcpp::attributes::SourceFileAttributes::hasInterface\(\)](#), `initFunctions_`, [Rcpp::attributes::Function::isHidden\(\)](#), [Rcpp::attributes::kInitAttribute](#), [Rcpp::attributes::kInterfaceCpp](#), [Rcpp::attributes::SourceFileAttributes::modules\(\)](#), `modules_`, `nativeRoutines_`, [Rcpp::attributes::ExportsGenerator::ostr\(\)](#), [Rcpp::attributes::ExportsGenerator::packageCppPrefix\(\)](#), [Rcpp::Rcout](#), [Rcpp::attributes::Function::renamedTo\(\)](#), and [Rcpp::attributes::SourceFileAttributes::sourceFile\(\)](#).

Here is the call graph for this function:



### 6.111.3.3 registerCCallable()

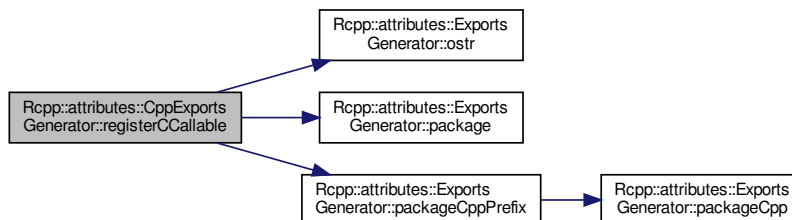
```
std::string Rcpp::attributes::CppExportsGenerator::registerCCallable (
    size_t indent,
    const std::string & exportedName,
    const std::string & name ) const [private]
```

Definition at line 2151 of file attributes.cpp.

References [Rcpp::attributes::ExportsGenerator::ostr\(\)](#), [Rcpp::attributes::ExportsGenerator::package\(\)](#), and [Rcpp::attributes::ExportsGenerator::packageCppPrefix\(\)](#).

Referenced by [writeEnd\(\)](#).

Here is the call graph for this function:



#### 6.111.3.4 writeBegin()

```
virtual void Rcpp::attributes::CppExportsGenerator::writeBegin ( ) [inline], [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 702 of file `attributes.cpp`.

#### 6.111.3.5 writeEnd()

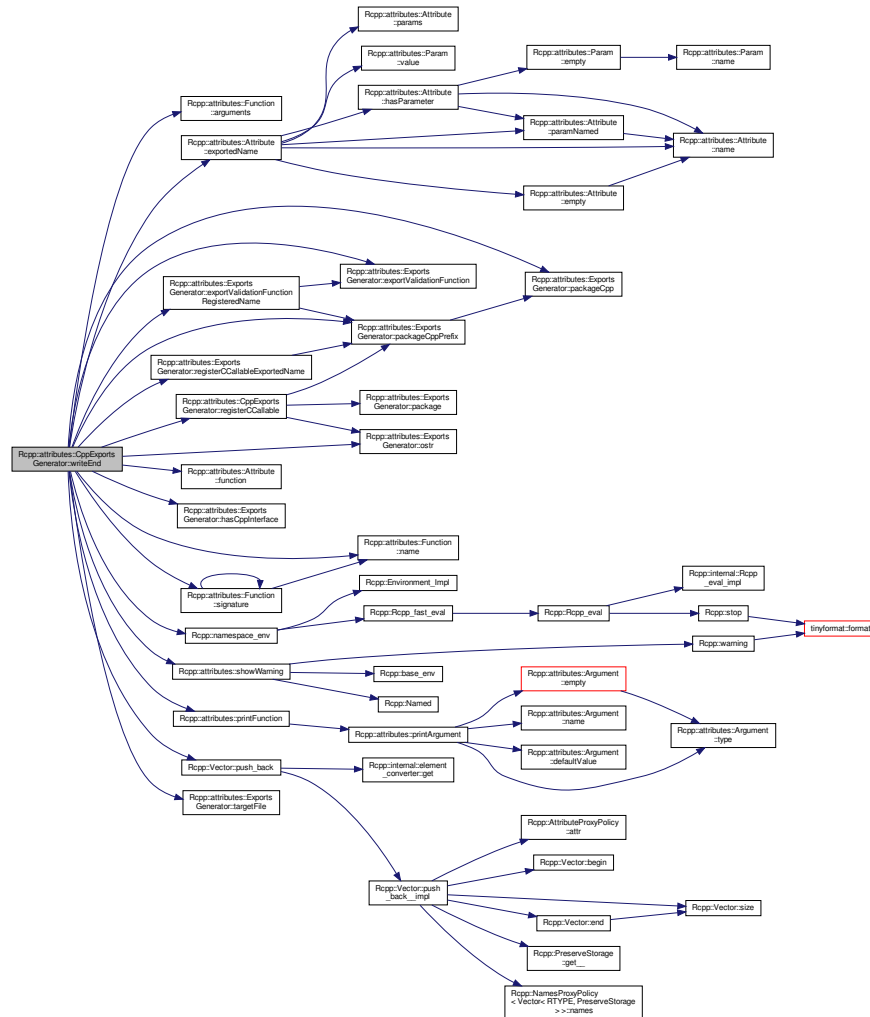
```
void Rcpp::attributes::CppExportsGenerator::writeEnd (
    bool hasPackageInit ) [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 2018 of file `attributes.cpp`.

References [Rcpp::attributes::Function::arguments\(\)](#), [callEntries](#), [cppExports\\_](#), [Rcpp::attributes::Attribute::exportedName\(\)](#), [Rcpp::attributes::ExportsGenerator::exportValidationFunction\(\)](#), [Rcpp::attributes::ExportsGenerator::exportValidationFunctionRegisteredName\(\)](#), [Rcpp::attributes::Attribute::function\(\)](#), [Rcpp::attributes::ExportsGenerator::hasCppInterface\(\)](#), [initFunctions\\_](#), [modules\\_](#), [Rcpp::attributes::Function::name\(\)](#), [Rcpp::namespace\\_env\(\)](#), [nativeRoutines\\_](#), [Rcpp::attributes::ExportsGenerator::ostr\(\)](#), [Rcpp::attributes::ExportsGenerator::packageCpp\(\)](#), [Rcpp::attributes::ExportsGenerator::packageCppPrefix\(\)](#), [Rcpp::attributes::printFunction\(\)](#), [Rcpp::Vector< RTYPE, StoragePolicy >::push\\_back\(\)](#), [registerCCallable\(\)](#), [Rcpp::attributes::ExportsGenerator::registerCCallableExportedName\(\)](#), [Rcpp::attributes::showWarning\(\)](#), [Rcpp::attributes::Function::signature\(\)](#), and [Rcpp::attributes::ExportsGenerator::targetFile\(\)](#).

Here is the call graph for this function:



## 6.111.4 Member Data Documentation

### 6.111.4.1 cppExports\_

```
std::vector<Attribute> Rcpp::attributes::CppExportsGenerator::cppExports_ [private]
```

Definition at line 719 of file `attributes.cpp`.

Referenced by `doWriteFunctions()`, and `writeEnd()`.

#### 6.111.4.2 `initFunctions_`

```
std::vector<Attribute> Rcpp::attributes::CppExportsGenerator::initFunctions_ [private]
```

Definition at line 716 of file `attributes.cpp`.

Referenced by `doWriteFunctions()`, and `writeEnd()`.

#### 6.111.4.3 `modules_`

```
std::vector<std::string> Rcpp::attributes::CppExportsGenerator::modules_ [private]
```

Definition at line 725 of file `attributes.cpp`.

Referenced by `doWriteFunctions()`, and `writeEnd()`.

#### 6.111.4.4 `nativeRoutines_`

```
std::vector<Attribute> Rcpp::attributes::CppExportsGenerator::nativeRoutines_ [private]
```

Definition at line 722 of file `attributes.cpp`.

Referenced by `doWriteFunctions()`, and `writeEnd()`.

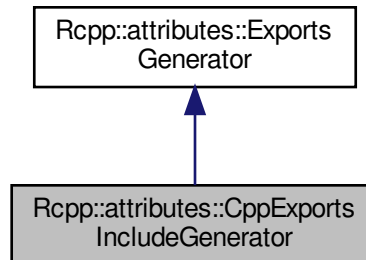
The documentation for this class was generated from the following file:

- [src/attributes.cpp](#)

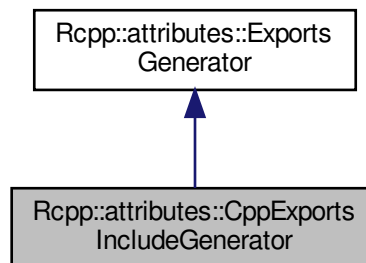


## 6.112 Rcpp::attributes::CppExportsIncludeGenerator Class Reference

Inheritance diagram for Rcpp::attributes::CppExportsIncludeGenerator:



Collaboration diagram for Rcpp::attributes::CppExportsIncludeGenerator:



### Public Member Functions

- [CppExportsIncludeGenerator](#) (const std::string &packageDir, const std::string &package, const std::string &fileSep)
- virtual void [writeBegin](#) ()
- virtual void [writeEnd](#) (bool hasPackageInit)
- virtual bool [commit](#) (const std::vector< std::string > &includes)

### Private Member Functions

- virtual void [doWriteFunctions](#) (const [SourceFileAttributes](#) &attributes, bool verbose)
- std::string [getCallable](#) (const std::string &function) const
- std::string [getHeaderGuard](#) () const

## Private Attributes

- `std::string` [includeDir\\_](#)

## Additional Inherited Members

### 6.112.1 Detailed Description

Definition at line 729 of file `attributes.cpp`.

### 6.112.2 Constructor & Destructor Documentation

#### 6.112.2.1 `CppExportsIncludeGenerator()`

```
Rcpp::attributes::CppExportsIncludeGenerator::CppExportsIncludeGenerator (
    const std::string & packageDir,
    const std::string & package,
    const std::string & fileSep )
```

Definition at line 2186 of file `attributes.cpp`.

References [includeDir\\_](#).

### 6.112.3 Member Function Documentation

#### 6.112.3.1 `commit()`

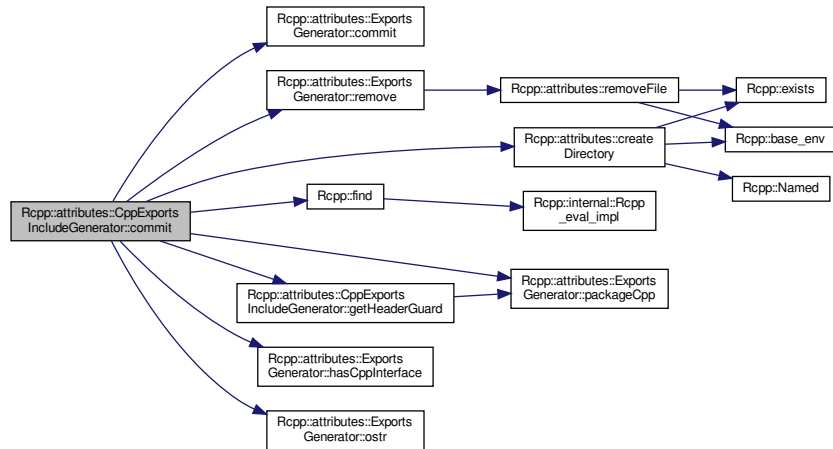
```
bool Rcpp::attributes::CppExportsIncludeGenerator::commit (
    const std::vector< std::string > & includes ) [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 2339 of file `attributes.cpp`.

References [Rcpp::attributes::ExportsGenerator::commit\(\)](#), [Rcpp::attributes::createDirectory\(\)](#), [Rcpp::find\(\)](#), [getHeaderGuard\(\)](#), [Rcpp::attributes::ExportsGenerator::hasCppInterface\(\)](#), [includeDir\\_](#), [Rcpp::attributes::ExportsGenerator::ostr\(\)](#), [Rcpp::attributes::ExportsGenerator::packageCpp\(\)](#), and [Rcpp::attributes::ExportsGenerator::remove\(\)](#).

Here is the call graph for this function:



### 6.112.3.2 doWriteFunctions()

```

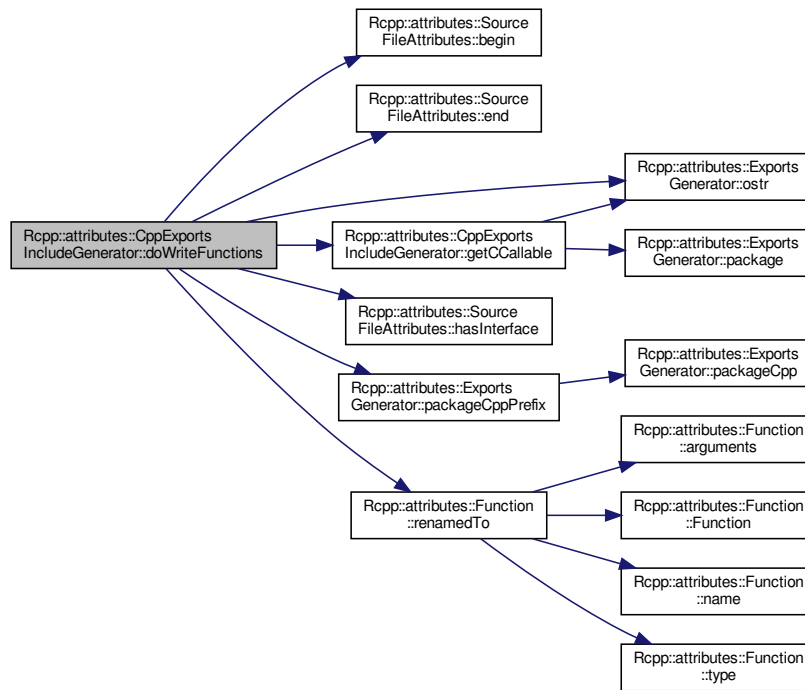
void Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions (
    const SourceFileAttributes & attributes,
    bool verbose ) [private], [virtual]
  
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 2248 of file `attributes.cpp`.

References [Rcpp::attributes::SourceFileAttributes::begin\(\)](#), [Rcpp::attributes::SourceFileAttributes::end\(\)](#), [get↔CCallable\(\)](#), [Rcpp::attributes::SourceFileAttributes::hasInterface\(\)](#), [Rcpp::attributes::kInterfaceCpp](#), [Rcpp::attributes↔::ExportsGenerator::ostr\(\)](#), [Rcpp::attributes::ExportsGenerator::packageCppPrefix\(\)](#), and [Rcpp::attributes::Function↔::renamedTo\(\)](#).

Here is the call graph for this function:



### 6.112.3.3 getCCallable()

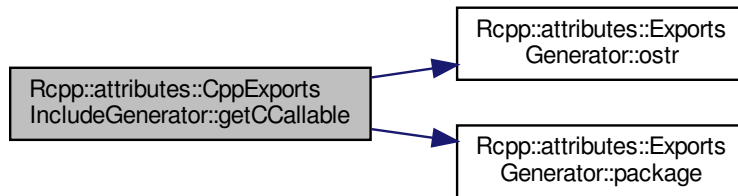
```
std::string Rcpp::attributes::CppExportsIncludeGenerator::getCCallable (
    const std::string & function ) const [private]
```

Definition at line 2391 of file `attributes.cpp`.

References `Rcpp::attributes::ExportsGenerator::ostr()`, and `Rcpp::attributes::ExportsGenerator::package()`.

Referenced by `doWriteFunctions()`, and `writeBegin()`.

Here is the call graph for this function:



#### 6.112.3.4 getHeaderGuard()

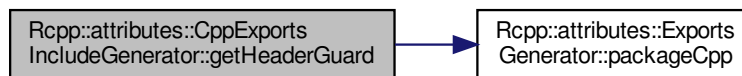
```
std::string Rcpp::attributes::CppExportsIncludeGenerator::getHeaderGuard ( ) const [private]
```

Definition at line 2400 of file `attributes.cpp`.

References `Rcpp::attributes::ExportsGenerator::packageCpp()`.

Referenced by `commit()`, and `writeEnd()`.

Here is the call graph for this function:



### 6.112.3.5 writeBegin()

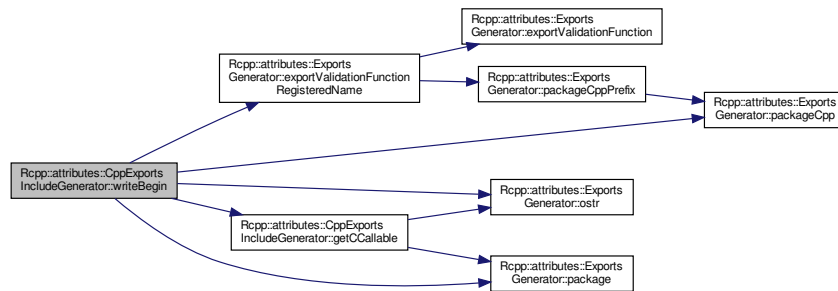
```
void Rcpp::attributes::CppExportsIncludeGenerator::writeBegin ( ) [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 2199 of file attributes.cpp.

References [Rcpp::attributes::ExportsGenerator::exportValidationFunctionRegisteredName\(\)](#), [getCCallable\(\)](#), [Rcpp::attributes::ExportsGenerator::ostr\(\)](#), [Rcpp::attributes::ExportsGenerator::package\(\)](#), and [Rcpp::attributes::ExportsGenerator::packageCpp\(\)](#).

Here is the call graph for this function:



### 6.112.3.6 writeEnd()

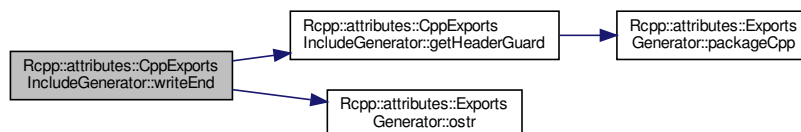
```
void Rcpp::attributes::CppExportsIncludeGenerator::writeEnd (
    bool hasPackageInit ) [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 2333 of file attributes.cpp.

References [getHeaderGuard\(\)](#), and [Rcpp::attributes::ExportsGenerator::ostr\(\)](#).

Here is the call graph for this function:



## 6.112.4 Member Data Documentation

### 6.112.4.1 includeDir\_

```
std::string Rcpp::attributes::CppExportsIncludeGenerator::includeDir_ [private]
```

Definition at line 746 of file attributes.cpp.

Referenced by commit(), and CppExportsIncludeGenerator().

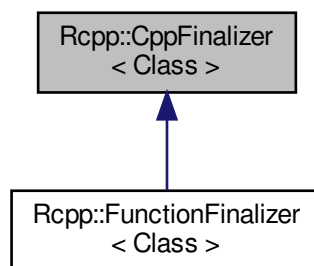
The documentation for this class was generated from the following file:

- [src/attributes.cpp](#)

## 6.113 Rcpp::CppFinalizer< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::CppFinalizer< Class >:



### Public Member Functions

- [CppFinalizer](#) ()
- virtual void [run](#) (Class \*)

### 6.113.1 Detailed Description

```
template<typename Class>  
class Rcpp::CppFinalizer< Class >
```

Definition at line 300 of file Module.h.

### 6.113.2 Constructor & Destructor Documentation

#### 6.113.2.1 CppFinalizer()

```
template<typename Class >  
Rcpp::CppFinalizer< Class >::CppFinalizer ( ) [inline]
```

Definition at line 302 of file Module.h.

### 6.113.3 Member Function Documentation

#### 6.113.3.1 run()

```
template<typename Class >  
virtual void Rcpp::CppFinalizer< Class >::run (  
    Class * ) [inline], [virtual]
```

Reimplemented in [Rcpp::FunctionFinalizer< Class >](#).

Definition at line 303 of file Module.h.

The documentation for this class was generated from the following file:

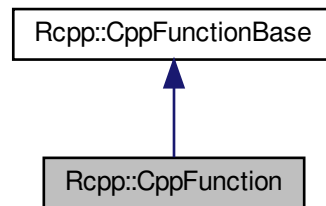
- [inst/include/Rcpp/Module.h](#)



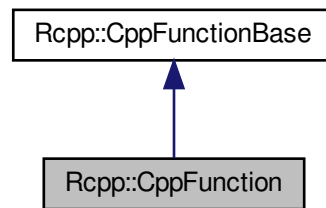
## 6.114 Rcpp::CppClass Class Reference

```
#include <CppClass.h>
```

Inheritance diagram for Rcpp::CppClass:



Collaboration diagram for Rcpp::CppClass:



### Public Member Functions

- [CppClass](#) (const char \*doc=0)
- virtual [~CppClass](#) ()
- virtual int [nargs](#) ()
- virtual bool [is\\_void](#) ()
- virtual void [signature](#) (std::string &, const char \*)
- virtual SEXP [get\\_formals](#) ()
- virtual DL\_FUNC [get\\_function\\_ptr](#) ()=0

### Public Attributes

- std::string [docstring](#)

### 6.114.1 Detailed Description

base class of all exported C++ functions. Template deduction in the `Module_generated_function.h` file creates an instance of a class that derives [CppClassFunction](#) (see `Module_generated_CppFunction.h` for all the definitions)

Definition at line 52 of file `CppClassFunction.h`.

### 6.114.2 Constructor & Destructor Documentation

#### 6.114.2.1 CppFunction()

```
Rcpp::CppClassFunction::CppClassFunction (
    const char * doc = 0 ) [inline]
```

Definition at line 54 of file `CppClassFunction.h`.

#### 6.114.2.2 ~CppClassFunction()

```
virtual Rcpp::CppClassFunction::~CppClassFunction ( ) [inline], [virtual]
```

Definition at line 55 of file `CppClassFunction.h`.

### 6.114.3 Member Function Documentation

#### 6.114.3.1 get\_formals()

```
virtual SEXP Rcpp::CppClassFunction::get_formals ( ) [inline], [virtual]
```

formal arguments

Definition at line 75 of file `CppClassFunction.h`.

Referenced by `Rcpp::Module::get_function()`.

### 6.114.3.2 get\_function\_ptr()

```
virtual DL_FUNC Rcpp::CppClass::get_function_ptr ( ) [pure virtual]
```

The actual function pointer, as a catch all function pointer (see Rdynload.h for definition of DL\_FUNC)

Referenced by Rcpp::Module::Add(), and Rcpp::Module::get\_function\_ptr().

### 6.114.3.3 is\_void()

```
virtual bool Rcpp::CppClass::is_void ( ) [inline], [virtual]
```

voidness

Definition at line 65 of file CppFunction.h.

Referenced by Rcpp::Module::get\_function(), and Rcpp::Module::invoke().

### 6.114.3.4 nargs()

```
virtual int Rcpp::CppClass::nargs ( ) [inline], [virtual]
```

The number of arguments of the function

Definition at line 60 of file CppFunction.h.

Referenced by Rcpp::Module::get\_function(), and Rcpp::Module::invoke().

### 6.114.3.5 signature()

```
virtual void Rcpp::CppClass::signature (
    std::string & ,
    const char * ) [inline], [virtual]
```

Human readable function signature (demangled if possible)

Definition at line 70 of file CppFunction.h.

Referenced by Rcpp::Module::get\_function().

## 6.114.4 Member Data Documentation

### 6.114.4.1 docstring

```
std::string Rcpp::CppMethod::docstring
```

description of the function

Definition at line 86 of file CppFunction.h.

Referenced by Rcpp::Module::get\_function().

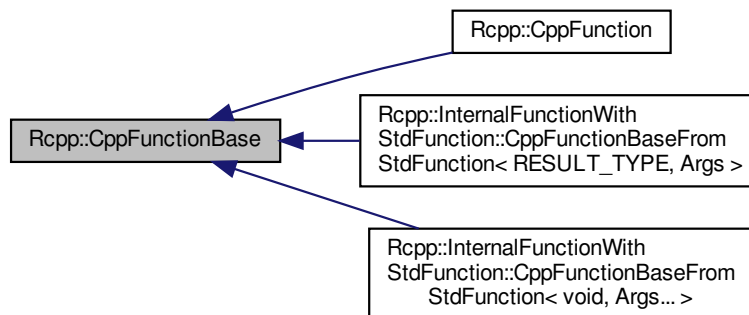
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/module/CppMethod.h](#)

## 6.115 Rcpp::CppMethodBase Class Reference

```
#include <CppMethod.h>
```

Inheritance diagram for Rcpp::CppMethodBase:



### Public Member Functions

- [CppMethodBase](#) ()
- virtual [~CppMethodBase](#) ()
- virtual SEXP [operator\(\)](#) (SEXP \*)

### 6.115.1 Detailed Description

base class for a callable function. This limited functionality is just barely enough for an InternalFunction, nothing more.

Definition at line 31 of file CppFunction.h.

### 6.115.2 Constructor & Destructor Documentation

#### 6.115.2.1 CppFunctionBase()

```
Rcpp::CppMethodBase::CppMethodBase ( ) [inline]
```

Definition at line 33 of file CppFunction.h.

#### 6.115.2.2 ~CppMethodBase()

```
virtual Rcpp::CppMethodBase::~~CppMethodBase ( ) [inline], [virtual]
```

Definition at line 34 of file CppFunction.h.

### 6.115.3 Member Function Documentation

#### 6.115.3.1 operator>()

```
virtual SEXP Rcpp::CppMethodBase::operator() (
    SEXP * ) [inline], [virtual]
```

modules call the function with this interface. input: an array of SEXP output: a SEXP.

Reimplemented in [Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< void, Args... >](#), and [Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< RESULT\\_TYPE, Args >](#).

Definition at line 40 of file CppFunction.h.

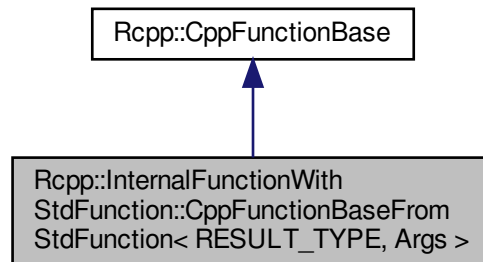
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/module/CppMethodBase.h](#)

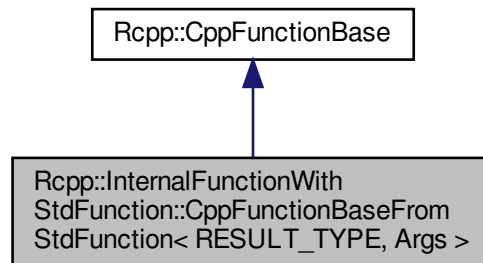
## 6.116 Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< RESULT\_TYPE, Args > Class Template Reference

```
#include <InternalFunctionWithStdFunction.h>
```

Inheritance diagram for Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< RESULT\_TYPE, Args >:



Collaboration diagram for Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< RESULT\_TYPE, Args >:



### Public Member Functions

- [CppMethodBaseFromStdFunction](#) (const std::function< RESULT\_TYPE(Args...) > &fun)
- virtual [~CppMethodBaseFromStdFunction](#) ()
- SEXP [operator\(\)](#) (SEXP \*args)

## Private Attributes

- `const std::function< RESULT_TYPE(Args...)>` [fun](#)

### 6.116.1 Detailed Description

```
template<typename RESULT_TYPE, typename... Args>  
class Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< RESULT_TYPE, Args >
```

Definition at line 35 of file InternalFunctionWithStdFunction.h.

### 6.116.2 Constructor & Destructor Documentation

#### 6.116.2.1 CppFunctionBaseFromStdFunction()

```
template<typename RESULT_TYPE , typename... Args>  
Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< RESULT_TYPE, Args >←  
::CppMethodBaseFromStdFunction (   
    const std::function< RESULT_TYPE(Args...)> & fun ) [inline]
```

Definition at line 37 of file InternalFunctionWithStdFunction.h.

#### 6.116.2.2 ~CppMethodBaseFromStdFunction()

```
template<typename RESULT_TYPE , typename... Args>  
virtual Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< RESULT_TYPE, Args  
>::~~CppMethodBaseFromStdFunction ( ) [inline], [virtual]
```

Definition at line 38 of file InternalFunctionWithStdFunction.h.

### 6.116.3 Member Function Documentation

### 6.116.3.1 operator()

```
template<typename RESULT_TYPE , typename... Args>
SEXP Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< RESULT_TYPE, Args >↵
::operator() (
    SEXP * ) [inline], [virtual]
```

modules call the function with this interface. input: an array of SEXP output: a SEXP.

Reimplemented from [Rcpp::CppMethodBase](#).

Definition at line 40 of file InternalFunctionWithStdFunction.h.

References `BEGIN_RCPP`, `END_RCPP`, and `Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< RESULT_TYPE, Args >::fun`.

## 6.116.4 Member Data Documentation

### 6.116.4.1 fun

```
template<typename RESULT_TYPE , typename... Args>
const std::function<RESULT_TYPE(Args...)> Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction<
RESULT_TYPE, Args >::fun [private]
```

Definition at line 48 of file InternalFunctionWithStdFunction.h.

Referenced by `Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< RESULT_TYPE, Args >::operator()`, and `Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< void, Args... >↵::operator()`.

The documentation for this class was generated from the following file:

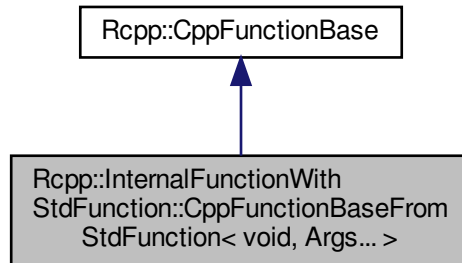
- `inst/include/Rcpp/InternalFunctionWithStdFunction.h`



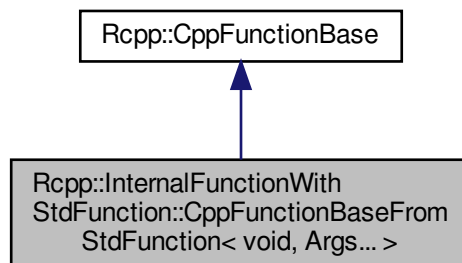
## 6.117 Rcpp::InternalFunctionWithStdFunction::CppClassBaseFromStdFunction< void, Args... > Class Template Reference

```
#include <InternalFunctionWithStdFunction.h>
```

Inheritance diagram for Rcpp::InternalFunctionWithStdFunction::CppClassBaseFromStdFunction< void, Args... >:



Collaboration diagram for Rcpp::InternalFunctionWithStdFunction::CppClassBaseFromStdFunction< void, Args... >:



### Public Member Functions

- [CppClassBaseFromStdFunction](#) (const std::function< void(Args...)> &fun)
- virtual [~CppClassBaseFromStdFunction](#) ()
- [SEXP operator\(\)](#) (SEXP \*args)

## Private Attributes

- `const std::function< void(Args...)>` [fun](#)

### 6.117.1 Detailed Description

```
template<typename... Args>
class Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >
```

Definition at line 52 of file InternalFunctionWithStdFunction.h.

### 6.117.2 Constructor & Destructor Documentation

#### 6.117.2.1 CppFunctionBaseFromStdFunction()

```
template<typename... Args>
Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >::CppFunctionBaseFromStdFunction(
    const std::function< void(Args...)> & fun ) [inline]
```

Definition at line 54 of file InternalFunctionWithStdFunction.h.

#### 6.117.2.2 ~CppFunctionBaseFromStdFunction()

```
template<typename... Args>
virtual Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >::~~CppFunctionBaseFromStdFunction ( ) [inline], [virtual]
```

Definition at line 55 of file InternalFunctionWithStdFunction.h.

### 6.117.3 Member Function Documentation

### 6.117.3.1 operator>()

```
template<typename... Args>
SEXP Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< void, Args... >←
::operator() (
    SEXP * ) [inline], [virtual]
```

modules call the function with this interface. input: an array of SEXP output: a SEXP.

Reimplemented from [Rcpp::CppMethodBase](#).

Definition at line 57 of file InternalFunctionWithStdFunction.h.

References `BEGIN_RCPP`, `END_RCPP`, and [Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction< RESULT\\_TYPE, Args >::fun](#).

## 6.117.4 Member Data Documentation

### 6.117.4.1 fun

```
template<typename... Args>
const std::function<void(Args...)> Rcpp::InternalFunctionWithStdFunction::CppMethodBaseFromStdFunction<
void, Args... >::fun [private]
```

Definition at line 64 of file InternalFunctionWithStdFunction.h.

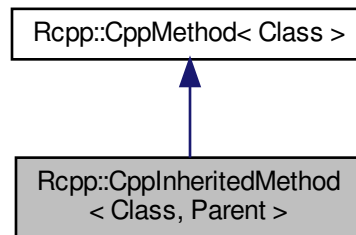
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/InternalFunctionWithStdFunction.h](#)

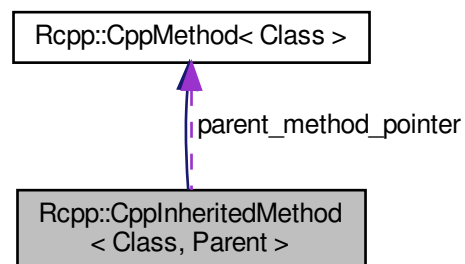
## 6.118 Rcpp::CppInheritedMethod< Class, Parent > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for `Rcpp::CppInheritedMethod< Class, Parent >`:



Collaboration diagram for `Rcpp::CppInheritedMethod< Class, Parent >`:



## Public Types

- typedef [Rcpp::XPtr< Class > XP](#)
- typedef [CppMethod< Parent > ParentMethod](#)

## Public Member Functions

- [CppInheritedMethod](#) ([ParentMethod](#) \*parent\_method\_pointer\_)
- `SEXP operator()` (`Class *object`, `SEXP *args`)
- `int nargs` ()
- `bool is_void` ()
- `bool is_const` ()
- `void signature` (`std::string &s`, `const char *name`)

## Private Attributes

- [ParentMethod](#) \* [parent\\_method\\_pointer](#)

### 6.118.1 Detailed Description

```
template<typename Class, typename Parent>
class Rcpp::CppInheritedMethod< Class, Parent >
```

Definition at line 112 of file Module.h.

### 6.118.2 Member Typedef Documentation

#### 6.118.2.1 ParentMethod

```
template<typename Class , typename Parent >
typedef CppMethod<Parent> Rcpp::CppInheritedMethod< Class, Parent >::ParentMethod
```

Definition at line 115 of file Module.h.

#### 6.118.2.2 XP

```
template<typename Class , typename Parent >
typedef Rcpp::XPtr<Class> Rcpp::CppInheritedMethod< Class, Parent >::XP
```

Definition at line 114 of file Module.h.

### 6.118.3 Constructor & Destructor Documentation

#### 6.118.3.1 CppInheritedMethod()

```
template<typename Class , typename Parent >
Rcpp::CppInheritedMethod< Class, Parent >::CppMethod (
    ParentMethod * parent\_method\_pointer\_ ) [inline]
```

Definition at line 117 of file Module.h.

## 6.118.4 Member Function Documentation

### 6.118.4.1 is\_const()

```
template<typename Class , typename Parent >
bool Rcpp::CppMethodInheritedMethod< Class, Parent >::is_const ( ) [inline], [virtual]
```

Reimplemented from [Rcpp::CppMethod< Class >](#).

Definition at line 126 of file Module.h.

References [Rcpp::CppMethod< Class >::is\\_const\(\)](#), and [Rcpp::CppMethodInheritedMethod< Class, Parent >::parent\\_↔method\\_pointer](#).

Here is the call graph for this function:



### 6.118.4.2 is\_void()

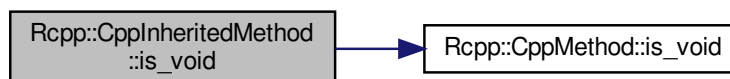
```
template<typename Class , typename Parent >
bool Rcpp::CppMethodInheritedMethod< Class, Parent >::is_void ( ) [inline], [virtual]
```

Reimplemented from [Rcpp::CppMethod< Class >](#).

Definition at line 125 of file Module.h.

References [Rcpp::CppMethod< Class >::is\\_void\(\)](#), and [Rcpp::CppMethodInheritedMethod< Class, Parent >::parent\\_↔method\\_pointer](#).

Here is the call graph for this function:



### 6.118.4.3 nargs()

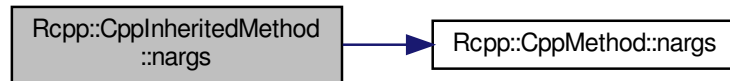
```
template<typename Class , typename Parent >  
int Rcpp::CppInheritedMethod< Class, Parent >::nargs ( ) [inline], [virtual]
```

Reimplemented from [Rcpp::CppMethod< Class >](#).

Definition at line 124 of file Module.h.

References [Rcpp::CppMethod< Class >::nargs\(\)](#), and [Rcpp::CppInheritedMethod< Class, Parent >::parent\\_method\\_pointer](#).

Here is the call graph for this function:



### 6.118.4.4 operator>()

```
template<typename Class , typename Parent >  
SEXP Rcpp::CppInheritedMethod< Class, Parent >::operator() (  
    Class * object,  
    SEXP * args ) [inline], [virtual]
```

Reimplemented from [Rcpp::CppMethod< Class >](#).

Definition at line 121 of file Module.h.

References [Rcpp::CppMethod< Class, Parent >::parent\\_method\\_pointer](#).

### 6.118.4.5 signature()

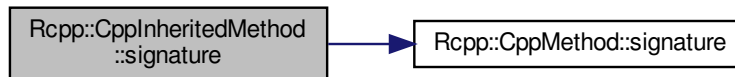
```
template<typename Class , typename Parent >
void Rcpp::CppMethod< Class, Parent >::signature (
    std::string & s,
    const char * name ) [inline], [virtual]
```

Reimplemented from [Rcpp::CppMethod< Class >](#).

Definition at line 127 of file Module.h.

References [Rcpp::CppMethod< Class, Parent >::parent\\_method\\_pointer](#), and [Rcpp::CppMethod< Class >::signature\(\)](#).

Here is the call graph for this function:



## 6.118.5 Member Data Documentation

### 6.118.5.1 parent\_method\_pointer

```
template<typename Class , typename Parent >
ParentMethod* Rcpp::CppMethod< Class, Parent >::parent_method_pointer [private]
```

Definition at line 130 of file Module.h.

Referenced by [Rcpp::CppMethod< Class, Parent >::is\\_const\(\)](#), [Rcpp::CppMethod< Class, Parent >::is\\_void\(\)](#), [Rcpp::CppMethod< Class, Parent >::nargs\(\)](#), [Rcpp::CppMethod< Class, Parent >::operator\(\)](#), and [Rcpp::CppMethod< Class, Parent >::signature\(\)](#).

The documentation for this class was generated from the following file:

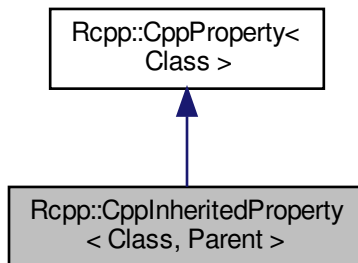
- [inst/include/Rcpp/Module.h](#)



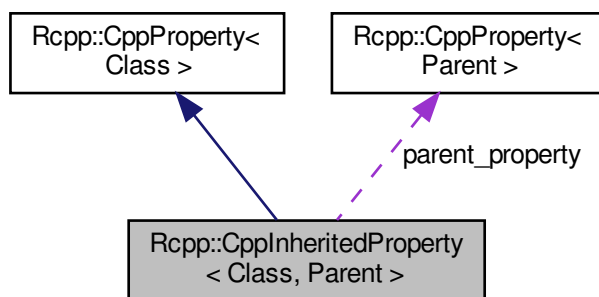
## 6.119 Rcpp::CppInheritedProperty< Class, Parent > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::CppInheritedProperty< Class, Parent >:



Collaboration diagram for Rcpp::CppInheritedProperty< Class, Parent >:



### Public Types

- typedef `CppProperty< Class >` `Base`

## Public Member Functions

- [CppInheritedProperty](#) ([CppProperty](#)< Parent > \*parent\_property\_)
- SEXP [get](#) (Class \*obj)
- void [set](#) (Class \*obj, SEXP s)
- bool [is\\_readonly](#) ()
- std::string [get\\_class](#) ()

## Private Attributes

- [CppProperty](#)< Parent > \* [parent\\_property](#)

## Additional Inherited Members

### 6.119.1 Detailed Description

```
template<typename Class, typename Parent>
class Rcpp::CppInheritedProperty < Class, Parent >
```

Definition at line 281 of file Module.h.

### 6.119.2 Member Typedef Documentation

#### 6.119.2.1 Base

```
template<typename Class , typename Parent >
typedef CppProperty<Class> Rcpp::CppInheritedProperty< Class, Parent >::Base
```

Definition at line 283 of file Module.h.

### 6.119.3 Constructor & Destructor Documentation

#### 6.119.3.1 CppInheritedProperty()

```
template<typename Class , typename Parent >
Rcpp::CppInheritedProperty< Class, Parent >::CppInheritedProperty (
    CppProperty< Parent > * parent_property_ ) [inline]
```

Definition at line 285 of file Module.h.

## 6.119.4 Member Function Documentation

### 6.119.4.1 get()

```
template<typename Class , typename Parent >
SEXP Rcpp::CppInheritedProperty< Class, Parent >::get (
    Class * obj ) [inline], [virtual]
```

Reimplemented from [Rcpp::CppProperty< Class >](#).

Definition at line 290 of file Module.h.

References [Rcpp::CppProperty< Class >::get\(\)](#), and [Rcpp::CppInheritedProperty< Class, Parent >::parent\\_property](#).

Here is the call graph for this function:



### 6.119.4.2 get\_class()

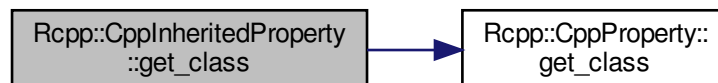
```
template<typename Class , typename Parent >
std::string Rcpp::CppInheritedProperty< Class, Parent >::get_class ( ) [inline], [virtual]
```

Reimplemented from [Rcpp::CppProperty< Class >](#).

Definition at line 293 of file Module.h.

References [Rcpp::CppProperty< Class >::get\\_class\(\)](#), and [Rcpp::CppInheritedProperty< Class, Parent >::parent\\_property](#).

Here is the call graph for this function:



### 6.119.4.3 is\_readonly()

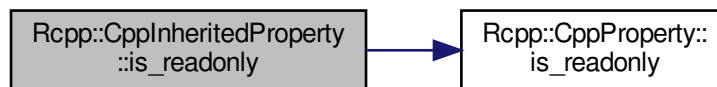
```
template<typename Class , typename Parent >
bool Rcpp::CppInheritedProperty< Class, Parent >::is_readonly ( ) [inline], [virtual]
```

Reimplemented from [Rcpp::CppClass< Class >](#).

Definition at line 292 of file Module.h.

References [Rcpp::CppClass< Class >::is\\_readonly\(\)](#), and [Rcpp::CppInheritedProperty< Class, Parent >::parent←\\_property](#).

Here is the call graph for this function:



### 6.119.4.4 set()

```
template<typename Class , typename Parent >
void Rcpp::CppInheritedProperty< Class, Parent >::set (
    Class * obj,
    SEXP s ) [inline], [virtual]
```

Reimplemented from [Rcpp::CppClass< Class >](#).

Definition at line 291 of file Module.h.

References [Rcpp::CppInheritedProperty< Class, Parent >::parent\\_property](#), and [Rcpp::CppClass< Class >::set\(\)](#).

Here is the call graph for this function:



## 6.119.5 Member Data Documentation

### 6.119.5.1 parent\_property

```
template<typename Class , typename Parent >
CppClassProperty<Parent>* Rcpp::CppClassInheritedProperty< Class, Parent >::parent_property [private]
```

Definition at line 296 of file Module.h.

Referenced by Rcpp::CppClassInheritedProperty< Class, Parent >::get(), Rcpp::CppClassInheritedProperty< Class, Parent >::get\_class(), Rcpp::CppClassInheritedProperty< Class, Parent >::is\_readonly(), and Rcpp::CppClassInheritedProperty< Class, Parent >::set().

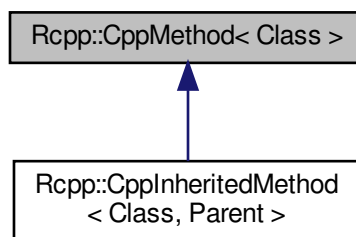
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

## 6.120 Rcpp::CppMethod< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::CppMethod< Class >:



### Public Types

- typedef `Rcpp::XPtr< Class > XP`

## Public Member Functions

- [CppMethod](#) ()
- virtual SEXP [operator\(\)](#) (Class \*, SEXP \*)
- virtual [~CppMethod](#) ()
- virtual int [nargs](#) ()
- virtual bool [is\\_void](#) ()
- virtual bool [is\\_const](#) ()
- virtual void [signature](#) (std::string &s, const char \*name)

### 6.120.1 Detailed Description

```
template<typename Class>  
class Rcpp::CppMethod< Class >
```

Definition at line 98 of file Module.h.

### 6.120.2 Member Typedef Documentation

#### 6.120.2.1 XP

```
template<typename Class >  
typedef Rcpp::XPtr<Class> Rcpp::CppMethod< Class >::XP
```

Definition at line 100 of file Module.h.

### 6.120.3 Constructor & Destructor Documentation

#### 6.120.3.1 CppMethod()

```
template<typename Class >  
Rcpp::CppMethod< Class >::CppMethod ( ) [inline]
```

Definition at line 102 of file Module.h.

### 6.120.3.2 ~CppMethod()

```
template<typename Class >
virtual Rcpp::CppMethod< Class >::~CppMethod ( ) [inline], [virtual]
```

Definition at line 104 of file Module.h.

## 6.120.4 Member Function Documentation

### 6.120.4.1 is\_const()

```
template<typename Class >
virtual bool Rcpp::CppMethod< Class >::is_const ( ) [inline], [virtual]
```

Reimplemented in [Rcpp::CplInheritedMethod< Class, Parent >](#).

Definition at line 107 of file Module.h.

Referenced by [Rcpp::CplInheritedMethod< Class, Parent >::is\\_const\(\)](#), and [Rcpp::SignedMethod< Class >::is\\_↔const\(\)](#).

### 6.120.4.2 is\_void()

```
template<typename Class >
virtual bool Rcpp::CppMethod< Class >::is_void ( ) [inline], [virtual]
```

Reimplemented in [Rcpp::CplInheritedMethod< Class, Parent >](#).

Definition at line 106 of file Module.h.

Referenced by [Rcpp::CplInheritedMethod< Class, Parent >::is\\_void\(\)](#), and [Rcpp::SignedMethod< Class >::is\\_void\(\)](#).

### 6.120.4.3 nargs()

```
template<typename Class >
virtual int Rcpp::CppMethod< Class >::nargs ( ) [inline], [virtual]
```

Reimplemented in [Rcpp::CplInheritedMethod< Class, Parent >](#).

Definition at line 105 of file Module.h.

Referenced by [Rcpp::CplInheritedMethod< Class, Parent >::nargs\(\)](#), and [Rcpp::SignedMethod< Class >::nargs\(\)](#).

#### 6.120.4.4 operator()

```
template<typename Class >
virtual SEXP Rcpp::CppMethod< Class >::operator() (
    Class * ,
    SEXP * ) [inline], [virtual]
```

Reimplemented in [Rcpp::CplInheritedMethod< Class, Parent >](#).

Definition at line 103 of file Module.h.

#### 6.120.4.5 signature()

```
template<typename Class >
virtual void Rcpp::CppMethod< Class >::signature (
    std::string & s,
    const char * name ) [inline], [virtual]
```

Reimplemented in [Rcpp::CplInheritedMethod< Class, Parent >](#).

Definition at line 108 of file Module.h.

Referenced by [Rcpp::CplInheritedMethod< Class, Parent >::signature\(\)](#), and [Rcpp::SignedMethod< Class >::signature\(\)](#).

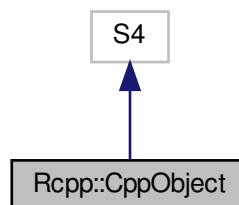
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Module.h](#)

## 6.121 Rcpp::CppObject Class Reference

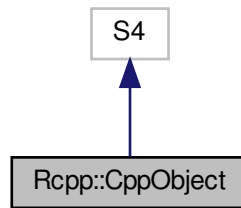
```
#include <Module.h>
```

Inheritance diagram for Rcpp::CppObject:





Collaboration diagram for Rcpp::CppObject:



## Public Types

- typedef [Rcpp::XPtr](#)< [Rcpp::Module](#) > XP

## Public Member Functions

- [CppObject](#) ([Module](#) \*p, [class\\_Base](#) \*clazz, SEXP xp)

## Private Types

- typedef [S4](#) Base

### 6.121.1 Detailed Description

Definition at line 412 of file Module.h.

### 6.121.2 Member Typedef Documentation

#### 6.121.2.1 Base

```
typedef S4 Rcpp::CppObject::Base [private]
```

Definition at line 413 of file Module.h.

### 6.121.2.2 XP

```
typedef Rcpp::XPtr<Rcpp::Module> Rcpp::CppObject::XP
```

Definition at line 415 of file Module.h.

## 6.121.3 Constructor & Destructor Documentation

### 6.121.3.1 CppObject()

```
Rcpp::CppObject::CppObject (
    Module * p,
    class_Base * clazz,
    SEXP xp ) [inline]
```

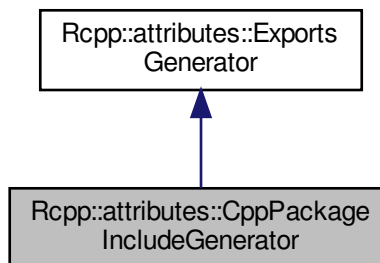
Definition at line 416 of file Module.h.

The documentation for this class was generated from the following file:

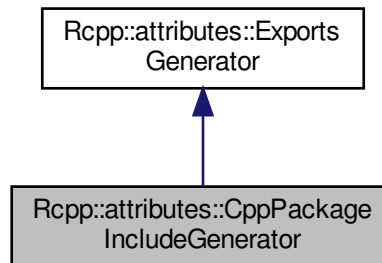
- inst/include/Rcpp/Module.h

## 6.122 Rcpp::attributes::CppPackageIncludeGenerator Class Reference

Inheritance diagram for Rcpp::attributes::CppPackageIncludeGenerator:



Collaboration diagram for Rcpp::attributes::CppPackageIncludeGenerator:



## Public Member Functions

- [CppPackageIncludeGenerator](#) (const std::string &packageDir, const std::string &package, const std::string &file↔ Sep)
- virtual void [writeBegin](#) ()
- virtual void [writeEnd](#) (bool hasPackageInit)
- virtual bool [commit](#) (const std::vector< std::string > &includes)

## Private Member Functions

- virtual void [doWriteFunctions](#) (const [SourceFileAttributes](#) &, bool)
- std::string [getHeaderGuard](#) () const

## Private Attributes

- std::string [includeDir\\_](#)

## Additional Inherited Members

### 6.122.1 Detailed Description

Definition at line 750 of file attributes.cpp.

### 6.122.2 Constructor & Destructor Documentation

### 6.122.2.1 CppPackageIncludeGenerator()

```
Rcpp::attributes::CppPackageIncludeGenerator::CppPackageIncludeGenerator (
    const std::string & packageDir,
    const std::string & package,
    const std::string & fileSep )
```

Definition at line 2404 of file attributes.cpp.

References includeDir\_.

## 6.122.3 Member Function Documentation

### 6.122.3.1 commit()

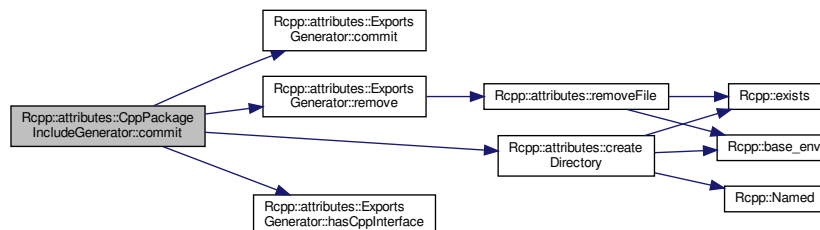
```
bool Rcpp::attributes::CppPackageIncludeGenerator::commit (
    const std::vector< std::string > & includes ) [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 2431 of file attributes.cpp.

References [Rcpp::attributes::ExportsGenerator::commit\(\)](#), [Rcpp::attributes::createDirectory\(\)](#), [Rcpp::attributes::ExportsGenerator::hasCppInterface\(\)](#), [includeDir\\_](#), and [Rcpp::attributes::ExportsGenerator::remove\(\)](#).

Here is the call graph for this function:



### 6.122.3.2 doWriteFunctions()

```
virtual void Rcpp::attributes::CppPackageIncludeGenerator::doWriteFunctions (
    const SourceFileAttributes & ,
    bool ) [inline], [private], [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 761 of file attributes.cpp.

### 6.122.3.3 getHeaderGuard()

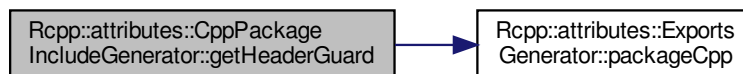
```
std::string Rcpp::attributes::CppPackageIncludeGenerator::getHeaderGuard ( ) const [private]
```

Definition at line 2445 of file attributes.cpp.

References [Rcpp::attributes::ExportsGenerator::packageCpp\(\)](#).

Referenced by [writeEnd\(\)](#).

Here is the call graph for this function:



### 6.122.3.4 writeBegin()

```
virtual void Rcpp::attributes::CppPackageIncludeGenerator::writeBegin ( ) [inline], [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 756 of file attributes.cpp.

### 6.122.3.5 writeEnd()

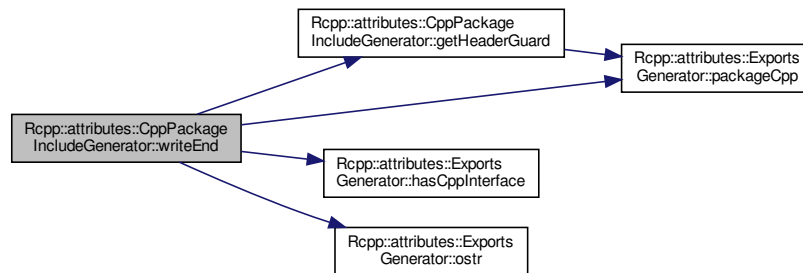
```
void Rcpp::attributes::CppPackageIncludeGenerator::writeEnd (
    bool hasPackageInit ) [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 2417 of file `attributes.cpp`.

References [getHeaderGuard\(\)](#), [Rcpp::attributes::ExportsGenerator::hasCplInterface\(\)](#), [Rcpp::attributes::ExportsGenerator::ostr\(\)](#), and [Rcpp::attributes::ExportsGenerator::packageCpp\(\)](#).

Here is the call graph for this function:



## 6.122.4 Member Data Documentation

### 6.122.4.1 includeDir\_

```
std::string Rcpp::attributes::CppPackageIncludeGenerator::includeDir_ [private]
```

Definition at line 765 of file `attributes.cpp`.

Referenced by [commit\(\)](#), and [CppPackageIncludeGenerator\(\)](#).

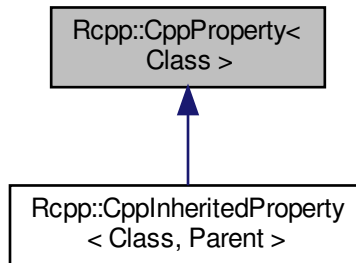
The documentation for this class was generated from the following file:

- [src/attributes.cpp](#)

## 6.123 Rcpp::CppClassProperty< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::CppClassProperty< Class >:



### Public Types

- typedef [Rcpp::XPtr< Class > XP](#)

### Public Member Functions

- [CppClassProperty](#) (const char \*doc=0)
- virtual [~CppClassProperty](#) ()
- virtual SEXP [get](#) (Class \*)
- virtual void [set](#) (Class \*, SEXP)
- virtual bool [is\\_readonly](#) ()
- virtual std::string [get\\_class](#) ()

### Public Attributes

- std::string [docstring](#)

### 6.123.1 Detailed Description

```
template<typename Class>  
class Rcpp::CppClassProperty< Class >
```

Definition at line 266 of file Module.h.

## 6.123.2 Member Typedef Documentation

### 6.123.2.1 XP

```
template<typename Class >
typedef Rcpp::XPtr<Class> Rcpp::CppClassProperty< Class >::XP
```

Definition at line 268 of file Module.h.

## 6.123.3 Constructor & Destructor Documentation

### 6.123.3.1 CppProperty()

```
template<typename Class >
Rcpp::CppClassProperty< Class >::CppClassProperty (
    const char * doc = 0 ) [inline]
```

Definition at line 270 of file Module.h.

### 6.123.3.2 ~CppClassProperty()

```
template<typename Class >
virtual Rcpp::CppClassProperty< Class >::~CppClassProperty ( ) [inline], [virtual]
```

Definition at line 271 of file Module.h.

## 6.123.4 Member Function Documentation

### 6.123.4.1 get()

```
template<typename Class >
virtual SEXP Rcpp::CppClassProperty< Class >::get (
    Class * ) [inline], [virtual]
```

Reimplemented in [Rcpp::CplInheritedProperty< Class, Parent >](#).

Definition at line 272 of file Module.h.

Referenced by [Rcpp::CplInheritedProperty< Class, Parent >::get\(\)](#).



#### 6.123.4.2 `get_class()`

```
template<typename Class >  
virtual std::string Rcpp::CppClass< Class >::get_class ( ) [inline], [virtual]
```

Reimplemented in [Rcpp::CplInheritedProperty< Class, Parent >](#).

Definition at line 275 of file Module.h.

Referenced by [Rcpp::CplInheritedProperty< Class, Parent >::get\\_class\(\)](#), and [Rcpp::S4\\_field< Class >::S4\\_field\(\)](#).

#### 6.123.4.3 `is_readonly()`

```
template<typename Class >  
virtual bool Rcpp::CppClass< Class >::is_readonly ( ) [inline], [virtual]
```

Reimplemented in [Rcpp::CplInheritedProperty< Class, Parent >](#).

Definition at line 274 of file Module.h.

Referenced by [Rcpp::CplInheritedProperty< Class, Parent >::is\\_readonly\(\)](#), and [Rcpp::S4\\_field< Class >::S4\\_field\(\)](#).

#### 6.123.4.4 `set()`

```
template<typename Class >  
virtual void Rcpp::CppClass< Class >::set (  
    Class * ,  
    SEXP ) [inline], [virtual]
```

Reimplemented in [Rcpp::CplInheritedProperty< Class, Parent >](#).

Definition at line 273 of file Module.h.

Referenced by [Rcpp::CplInheritedProperty< Class, Parent >::set\(\)](#).

### 6.123.5 Member Data Documentation

### 6.123.5.1 docstring

```
template<typename Class >
std::string Rcpp::CppClassProperty< Class >::docstring
```

Definition at line 277 of file Module.h.

Referenced by Rcpp::S4\_field< Class >::S4\_field().

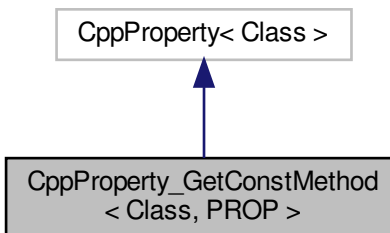
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

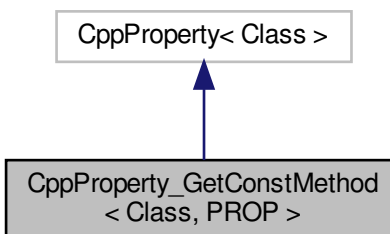
## 6.124 CppProperty\_GetConstMethod< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty\_GetConstMethod< Class, PROP >:



Collaboration diagram for CppProperty\_GetConstMethod< Class, PROP >:



## Public Types

- typedef PROP(Class::\* [GetMethod](#)) (void) const
- typedef CppProperty< Class > [prop\\_class](#)

## Public Member Functions

- [CppProperty\\_GetConstMethod](#) ([GetMethod](#) getter\_, const char \*doc=0)
- SEXP [get](#) (Class \*object)
- void [set](#) (Class \*, SEXP)
- bool [is\\_readonly](#) ()
- std::string [get\\_class](#) ()

## Private Attributes

- [GetMethod](#) [getter](#)
- std::string [class\\_name](#)

### 6.124.1 Detailed Description

```
template<typename Class, typename PROP>
class CppProperty_GetConstMethod< Class, PROP >
```

Definition at line 48 of file Module\_Property.h.

### 6.124.2 Member Typedef Documentation

#### 6.124.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(Class::* CppProperty\_GetConstMethod< Class, PROP >::GetMethod) (void) const
```

Definition at line 50 of file Module\_Property.h.

#### 6.124.2.2 prop\_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty\_GetConstMethod< Class, PROP >::prop_class
```

Definition at line 51 of file Module\_Property.h.

## 6.124.3 Constructor & Destructor Documentation

### 6.124.3.1 CppProperty\_GetConstMethod()

```
template<typename Class , typename PROP >
CppProperty_GetConstMethod< Class, PROP >::CppMethod (
    GetMethod getter_,
    const char * doc = 0 ) [inline]
```

Definition at line 53 of file Module\_Property.h.

## 6.124.4 Member Function Documentation

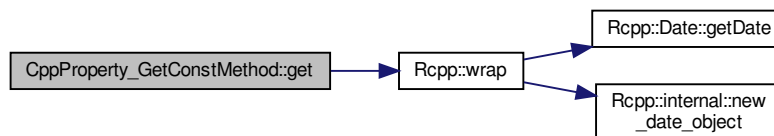
### 6.124.4.1 get()

```
template<typename Class , typename PROP >
SEXP CppProperty_GetConstMethod< Class, PROP >::get (
    Class * object ) [inline]
```

Definition at line 56 of file Module\_Property.h.

References CppProperty\_GetConstMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



#### 6.124.4.2 get\_class()

```
template<typename Class , typename PROP >  
std::string CppProperty_GetConstMethod< Class, PROP >::get_class ( ) [inline]
```

Definition at line 59 of file Module\_Property.h.

References CppProperty\_GetConstMethod< Class, PROP >::class\_name.

#### 6.124.4.3 is\_readonly()

```
template<typename Class , typename PROP >  
bool CppProperty_GetConstMethod< Class, PROP >::is_readonly ( ) [inline]
```

Definition at line 58 of file Module\_Property.h.

#### 6.124.4.4 set()

```
template<typename Class , typename PROP >  
void CppProperty_GetConstMethod< Class, PROP >::set (  
    Class * ,  
    SEXP ) [inline]
```

Definition at line 57 of file Module\_Property.h.

### 6.124.5 Member Data Documentation

#### 6.124.5.1 class\_name

```
template<typename Class , typename PROP >  
std::string CppProperty_GetConstMethod< Class, PROP >::class_name [private]
```

Definition at line 63 of file Module\_Property.h.

Referenced by CppProperty\_GetConstMethod< Class, PROP >::get\_class().

### 6.124.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetConstMethod< Class, PROP >::getter [private]
```

Definition at line 62 of file Module\_Property.h.

Referenced by CppProperty\_GetConstMethod< Class, PROP >::get().

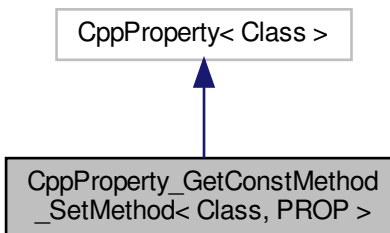
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/module/Module\\_Property.h](#)

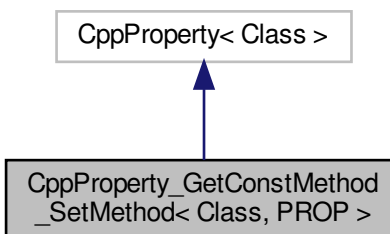
## 6.125 CppProperty\_GetConstMethod\_SetMethod< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty\_GetConstMethod\_SetMethod< Class, PROP >:



Collaboration diagram for CppProperty\_GetConstMethod\_SetMethod< Class, PROP >:



## Public Types

- typedef PROP(Class::\* [GetMethod](#)) (void) const
- typedef void(Class::\* [SetMethod](#)) (PROP)
- typedef CppProperty< Class > [prop\\_class](#)

## Public Member Functions

- [CppProperty\\_GetConstMethod\\_SetMethod](#) ([GetMethod](#) getter\_, [SetMethod](#) setter\_, const char \*doc=0)
- SEXP [get](#) (Class \*object)
- void [set](#) (Class \*object, SEXP value)
- bool [is\\_readonly](#) ()
- std::string [get\\_class](#) ()

## Private Attributes

- [GetMethod](#) [getter](#)
- [SetMethod](#) [setter](#)
- std::string [class\\_name](#)

### 6.125.1 Detailed Description

```
template<typename Class, typename PROP>
class CppProperty_GetConstMethod_SetMethod< Class, PROP >
```

Definition at line 115 of file Module\_Property.h.

### 6.125.2 Member Typedef Documentation

#### 6.125.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(Class::* CppProperty\_GetConstMethod\_SetMethod< Class, PROP >::GetMethod) (void)
const
```

Definition at line 117 of file Module\_Property.h.

### 6.125.2.2 prop\_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetConstMethod_SetMethod< Class, PROP >::prop_class
```

Definition at line 119 of file Module\_Property.h.

### 6.125.2.3 SetMethod

```
template<typename Class , typename PROP >
typedef void(Class::* CppProperty_GetConstMethod_SetMethod< Class, PROP >::SetMethod) (PROP)
```

Definition at line 118 of file Module\_Property.h.

## 6.125.3 Constructor & Destructor Documentation

### 6.125.3.1 CppProperty\_GetConstMethod\_SetMethod()

```
template<typename Class , typename PROP >
CppProperty_GetConstMethod_SetMethod< Class, PROP >::CppProperty_GetConstMethod_SetMethod (
    GetMethod getter_,
    SetMethod setter_,
    const char * doc = 0 ) [inline]
```

Definition at line 121 of file Module\_Property.h.

## 6.125.4 Member Function Documentation



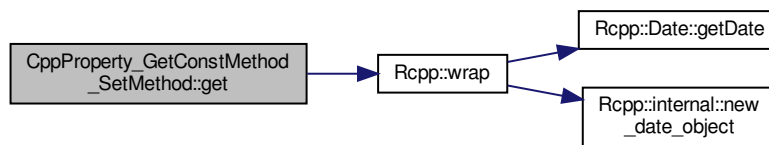
### 6.125.4.1 get()

```
template<typename Class , typename PROP >
SEXPR CppProperty_GetConstMethod_SetMethod< Class, PROP >::get (
    Class * object ) [inline]
```

Definition at line 124 of file Module\_Property.h.

References CppProperty\_GetConstMethod\_SetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



### 6.125.4.2 get\_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetConstMethod_SetMethod< Class, PROP >::get_class ( ) [inline]
```

Definition at line 131 of file Module\_Property.h.

References CppProperty\_GetConstMethod\_SetMethod< Class, PROP >::class\_name.

### 6.125.4.3 is\_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetConstMethod_SetMethod< Class, PROP >::is_readonly ( ) [inline]
```

Definition at line 130 of file Module\_Property.h.

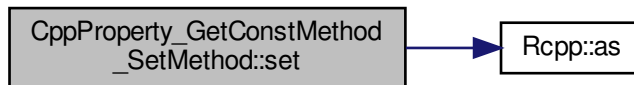
#### 6.125.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetConstMethod_SetMethod< Class, PROP >::set (
    Class * object,
    SEXP value ) [inline]
```

Definition at line 127 of file Module\_Property.h.

References Rcpp::as(), and CppProperty\_GetConstMethod\_SetMethod< Class, PROP >::setter.

Here is the call graph for this function:



### 6.125.5 Member Data Documentation

#### 6.125.5.1 class\_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetConstMethod_SetMethod< Class, PROP >::class_name [private]
```

Definition at line 136 of file Module\_Property.h.

Referenced by CppProperty\_GetConstMethod\_SetMethod< Class, PROP >::get\_class().

#### 6.125.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetConstMethod_SetMethod< Class, PROP >::getter [private]
```

Definition at line 134 of file Module\_Property.h.

Referenced by CppProperty\_GetConstMethod\_SetMethod< Class, PROP >::get().

## 6.125.5.3 setter

```
template<typename Class , typename PROP >
SetMethod CppProperty_GetConstMethod_SetMethod< Class, PROP >::setter [private]
```

Definition at line 135 of file Module\_Property.h.

Referenced by CppProperty\_GetConstMethod\_SetMethod< Class, PROP >::set().

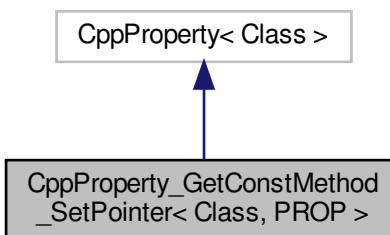
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module\_Property.h

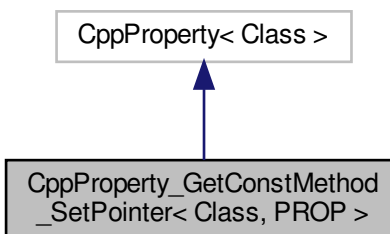
## 6.126 CppProperty\_GetConstMethod\_SetPointer< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty\_GetConstMethod\_SetPointer< Class, PROP >:



Collaboration diagram for CppProperty\_GetConstMethod\_SetPointer< Class, PROP >:



## Public Types

- typedef PROP(Class::\* [GetMethod](#)) (void) const
- typedef void(\* [SetMethod](#)) (Class \*, PROP)
- typedef CppProperty< Class > [prop\\_class](#)

## Public Member Functions

- [CppProperty\\_GetConstMethod\\_SetPointer](#) ([GetMethod](#) getter\_, [SetMethod](#) setter\_, const char \*doc=0)
- SEXP [get](#) (Class \*object)
- void [set](#) (Class \*object, SEXP value)
- bool [is\\_readonly](#) ()
- std::string [get\\_class](#) ()

## Private Attributes

- [GetMethod](#) [getter](#)
- [SetMethod](#) [setter](#)
- std::string [class\\_name](#)

### 6.126.1 Detailed Description

```
template<typename Class, typename PROP>
class CppProperty_GetConstMethod_SetPointer< Class, PROP >
```

Definition at line 170 of file Module\_Property.h.

### 6.126.2 Member Typedef Documentation

#### 6.126.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(Class::* CppProperty\_GetConstMethod\_SetPointer< Class, PROP >::GetMethod) (void)
const
```

Definition at line 172 of file Module\_Property.h.

### 6.126.2.2 prop\_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetConstMethod_SetPointer< Class, PROP >::prop_class
```

Definition at line 174 of file Module\_Property.h.

### 6.126.2.3 SetMethod

```
template<typename Class , typename PROP >
typedef void(* CppProperty_GetConstMethod_SetPointer< Class, PROP >::SetMethod) (Class *, PROP)
```

Definition at line 173 of file Module\_Property.h.

## 6.126.3 Constructor & Destructor Documentation

### 6.126.3.1 CppProperty\_GetConstMethod\_SetPointer()

```
template<typename Class , typename PROP >
CppProperty_GetConstMethod_SetPointer< Class, PROP >::CppProperty_GetConstMethod_SetPointer (
    GetMethod getter_,
    SetMethod setter_,
    const char * doc = 0 ) [inline]
```

Definition at line 176 of file Module\_Property.h.

## 6.126.4 Member Function Documentation

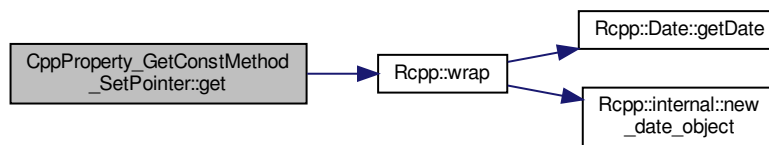
### 6.126.4.1 get()

```
template<typename Class , typename PROP >
SEXP CppProperty_GetConstMethod_SetPointer< Class, PROP >::get (
    Class * object ) [inline]
```

Definition at line 179 of file Module\_Property.h.

References CppProperty\_GetConstMethod\_SetPointer< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



### 6.126.4.2 get\_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetConstMethod_SetPointer< Class, PROP >::get_class ( ) [inline]
```

Definition at line 186 of file Module\_Property.h.

References CppProperty\_GetConstMethod\_SetPointer< Class, PROP >::class\_name.

### 6.126.4.3 is\_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetConstMethod_SetPointer< Class, PROP >::is_readonly ( ) [inline]
```

Definition at line 185 of file Module\_Property.h.

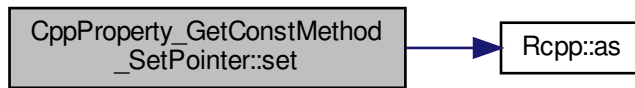
#### 6.126.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetConstMethod_SetPointer< Class, PROP >::set (
    Class * object,
    SEXP value ) [inline]
```

Definition at line 182 of file Module\_Property.h.

References Rcpp::as(), and CppProperty\_GetConstMethod\_SetPointer< Class, PROP >::setter.

Here is the call graph for this function:



### 6.126.5 Member Data Documentation

#### 6.126.5.1 class\_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetConstMethod_SetPointer< Class, PROP >::class_name [private]
```

Definition at line 191 of file Module\_Property.h.

Referenced by CppProperty\_GetConstMethod\_SetPointer< Class, PROP >::get\_class().

#### 6.126.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetConstMethod_SetPointer< Class, PROP >::getter [private]
```

Definition at line 189 of file Module\_Property.h.

Referenced by CppProperty\_GetConstMethod\_SetPointer< Class, PROP >::get().

### 6.126.5.3 setter

```
template<typename Class , typename PROP >
SetMethod CppProperty_GetConstMethod_SetPointer< Class, PROP >::setter [private]
```

Definition at line 190 of file Module\_Property.h.

Referenced by CppProperty\_GetConstMethod\_SetPointer< Class, PROP >::set().

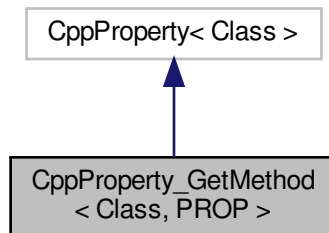
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/module/Module\\_Property.h](#)

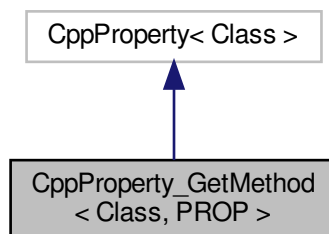
## 6.127 CppProperty\_GetMethod< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty\_GetMethod< Class, PROP >:



Collaboration diagram for CppProperty\_GetMethod< Class, PROP >:





## Public Types

- typedef PROP(Class::\* [GetMethod](#)) (void)
- typedef CppProperty< Class > [prop\\_class](#)

## Public Member Functions

- [CppProperty\\_GetMethod](#) ([GetMethod](#) getter\_, const char \*doc=0)
- SEXP [get](#) (Class \*object)
- void [set](#) (Class \*, SEXP)
- bool [is\\_readonly](#) ()
- std::string [get\\_class](#) ()

## Private Attributes

- [GetMethod](#) [getter](#)
- std::string [class\\_name](#)

### 6.127.1 Detailed Description

```
template<typename Class, typename PROP>
class CppProperty_GetMethod< Class, PROP >
```

Definition at line 27 of file Module\_Property.h.

### 6.127.2 Member Typedef Documentation

#### 6.127.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(Class::* CppProperty\_GetMethod< Class, PROP >::GetMethod) (void)
```

Definition at line 29 of file Module\_Property.h.

#### 6.127.2.2 prop\_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty\_GetMethod< Class, PROP >::prop_class
```

Definition at line 30 of file Module\_Property.h.

## 6.127.3 Constructor & Destructor Documentation

### 6.127.3.1 CppProperty\_GetMethod()

```
template<typename Class , typename PROP >
CppProperty_GetMethod< Class, PROP >::CppMethod_GetMethod (
    GetMethod getter_,
    const char * doc = 0 ) [inline]
```

Definition at line 32 of file Module\_Property.h.

## 6.127.4 Member Function Documentation

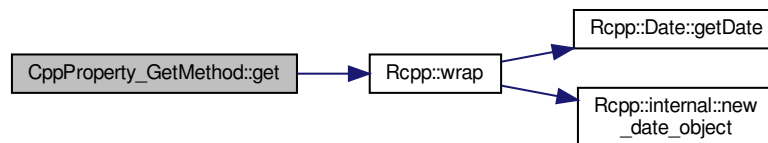
### 6.127.4.1 get()

```
template<typename Class , typename PROP >
SEXP CppProperty_GetMethod< Class, PROP >::get (
    Class * object ) [inline]
```

Definition at line 35 of file Module\_Property.h.

References CppProperty\_GetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



#### 6.127.4.2 get\_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetMethod< Class, PROP >::get_class ( ) [inline]
```

Definition at line 38 of file Module\_Property.h.

References CppProperty\_GetMethod< Class, PROP >::class\_name.

#### 6.127.4.3 is\_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetMethod< Class, PROP >::is_readonly ( ) [inline]
```

Definition at line 37 of file Module\_Property.h.

#### 6.127.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetMethod< Class, PROP >::set (
    Class * ,
    SEXP ) [inline]
```

Definition at line 36 of file Module\_Property.h.

### 6.127.5 Member Data Documentation

#### 6.127.5.1 class\_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetMethod< Class, PROP >::class_name [private]
```

Definition at line 42 of file Module\_Property.h.

Referenced by CppProperty\_GetMethod< Class, PROP >::get\_class().

### 6.127.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetMethod< Class, PROP >::getter [private]
```

Definition at line 41 of file Module\_Property.h.

Referenced by CppProperty\_GetMethod< Class, PROP >::get().

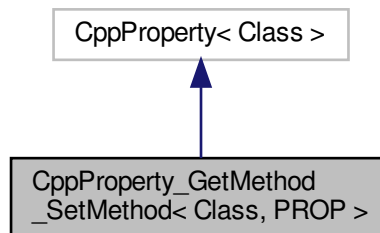
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/module/Module\\_Property.h](#)

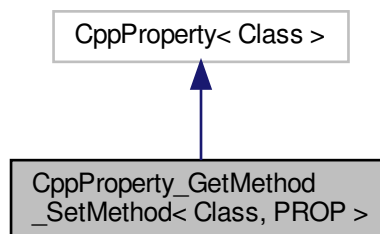
## 6.128 CppProperty\_GetMethod\_SetMethod< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty\_GetMethod\_SetMethod< Class, PROP >:



Collaboration diagram for CppProperty\_GetMethod\_SetMethod< Class, PROP >:



## Public Types

- typedef PROP(Class::\* [GetMethod](#)) (void)
- typedef void(Class::\* [SetMethod](#)) (PROP)
- typedef CppProperty< Class > [prop\\_class](#)

## Public Member Functions

- [CppMethod\\_GetMethod\\_SetMethod](#) ([GetMethod](#) getter\_, [SetMethod](#) setter\_, const char \*doc=0)
- SEXP [get](#) (Class \*object)
- void [set](#) (Class \*object, SEXP value)
- bool [is\\_readonly](#) ()
- std::string [get\\_class](#) ()

## Private Attributes

- [GetMethod](#) [getter](#)
- [SetMethod](#) [setter](#)
- std::string [class\\_name](#)

### 6.128.1 Detailed Description

```
template<typename Class, typename PROP>
class CppProperty_GetMethod_SetMethod< Class, PROP >
```

Definition at line 91 of file Module\_Property.h.

### 6.128.2 Member Typedef Documentation

#### 6.128.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(Class::* CppMethod\_GetMethod\_SetMethod< Class, PROP >::GetMethod) (void)
```

Definition at line 93 of file Module\_Property.h.

### 6.128.2.2 prop\_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetMethod_SetMethod< Class, PROP >::prop_class
```

Definition at line 95 of file Module\_Property.h.

### 6.128.2.3 SetMethod

```
template<typename Class , typename PROP >
typedef void(Class::* CppProperty_GetMethod_SetMethod< Class, PROP >::SetMethod) (PROP)
```

Definition at line 94 of file Module\_Property.h.

## 6.128.3 Constructor & Destructor Documentation

### 6.128.3.1 CppProperty\_GetMethod\_SetMethod()

```
template<typename Class , typename PROP >
CppProperty_GetMethod_SetMethod< Class, PROP >::CppProperty_GetMethod_SetMethod (
    GetMethod getter_,
    SetMethod setter_,
    const char * doc = 0 ) [inline]
```

Definition at line 97 of file Module\_Property.h.

## 6.128.4 Member Function Documentation

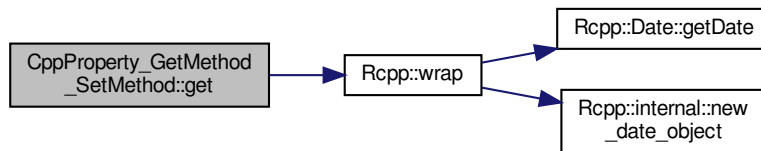
### 6.128.4.1 get()

```
template<typename Class , typename PROP >
SEXPR CppProperty_GetMethod_SetMethod< Class, PROP >::get (
    Class * object ) [inline]
```

Definition at line 100 of file Module\_Property.h.

References CppProperty\_GetMethod\_SetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



### 6.128.4.2 get\_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetMethod_SetMethod< Class, PROP >::get_class ( ) [inline]
```

Definition at line 107 of file Module\_Property.h.

References CppProperty\_GetMethod\_SetMethod< Class, PROP >::class\_name.

### 6.128.4.3 is\_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetMethod_SetMethod< Class, PROP >::is_readonly ( ) [inline]
```

Definition at line 106 of file Module\_Property.h.

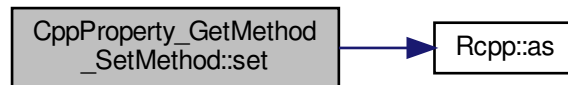
#### 6.128.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetMethod_SetMethod< Class, PROP >::set (
    Class * object,
    SEXP value ) [inline]
```

Definition at line 103 of file Module\_Property.h.

References Rcpp::as(), and CppProperty\_GetMethod\_SetMethod< Class, PROP >::setter.

Here is the call graph for this function:



### 6.128.5 Member Data Documentation

#### 6.128.5.1 class\_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetMethod_SetMethod< Class, PROP >::class_name [private]
```

Definition at line 112 of file Module\_Property.h.

Referenced by CppProperty\_GetMethod\_SetMethod< Class, PROP >::get\_class().

#### 6.128.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetMethod_SetMethod< Class, PROP >::getter [private]
```

Definition at line 110 of file Module\_Property.h.

Referenced by CppProperty\_GetMethod\_SetMethod< Class, PROP >::get().



### 6.128.5.3 setter

```
template<typename Class , typename PROP >  
SetMethod CppProperty_GetMethod_SetMethod< Class, PROP >::setter [private]
```

Definition at line 111 of file Module\_Property.h.

Referenced by CppProperty\_GetMethod\_SetMethod< Class, PROP >::set().

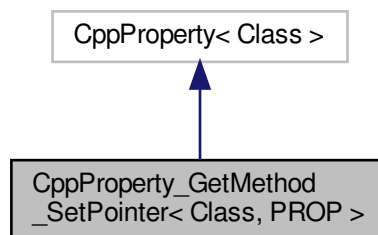
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/module/Module\\_Property.h](#)

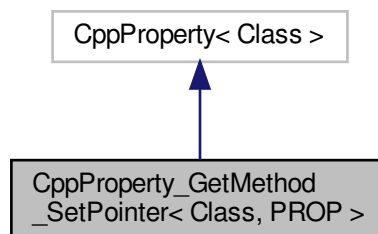
## 6.129 CppProperty\_GetMethod\_SetPointer< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty\_GetMethod\_SetPointer< Class, PROP >:



Collaboration diagram for CppProperty\_GetMethod\_SetPointer< Class, PROP >:



## Public Types

- typedef PROP(Class::\* [GetMethod](#)) (void)
- typedef void(\* [SetMethod](#)) (Class \*, PROP)
- typedef CppProperty< Class > [prop\\_class](#)

## Public Member Functions

- [CppProperty\\_GetMethod\\_SetPointer](#) ([GetMethod](#) getter\_, [SetMethod](#) setter\_, const char \*doc=0)
- SEXP [get](#) (Class \*object)
- void [set](#) (Class \*object, SEXP value)
- bool [is\\_readonly](#) ()
- std::string [get\\_class](#) ()

## Private Attributes

- [GetMethod](#) [getter](#)
- [SetMethod](#) [setter](#)
- std::string [class\\_name](#)

### 6.129.1 Detailed Description

```
template<typename Class, typename PROP>
class CppProperty_GetMethod_SetPointer< Class, PROP >
```

Definition at line 145 of file Module\_Property.h.

### 6.129.2 Member Typedef Documentation

#### 6.129.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(Class::* CppProperty\_GetMethod\_SetPointer< Class, PROP >::GetMethod) (void)
```

Definition at line 147 of file Module\_Property.h.

### 6.129.2.2 prop\_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetMethod_SetPointer< Class, PROP >::prop_class
```

Definition at line 149 of file Module\_Property.h.

### 6.129.2.3 SetMethod

```
template<typename Class , typename PROP >
typedef void(* CppProperty_GetMethod_SetPointer< Class, PROP >::SetMethod) (Class *, PROP)
```

Definition at line 148 of file Module\_Property.h.

## 6.129.3 Constructor & Destructor Documentation

### 6.129.3.1 CppProperty\_GetMethod\_SetPointer()

```
template<typename Class , typename PROP >
CppProperty_GetMethod_SetPointer< Class, PROP >::CppProperty_GetMethod_SetPointer (
    GetMethod getter_,
    SetMethod setter_,
    const char * doc = 0 ) [inline]
```

Definition at line 151 of file Module\_Property.h.

## 6.129.4 Member Function Documentation

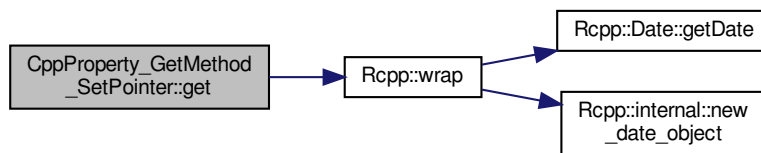
### 6.129.4.1 get()

```
template<typename Class , typename PROP >
SEXPR CppProperty_GetMethod_SetPointer< Class, PROP >::get (
    Class * object ) [inline]
```

Definition at line 154 of file Module\_Property.h.

References CppProperty\_GetMethod\_SetPointer< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



### 6.129.4.2 get\_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetMethod_SetPointer< Class, PROP >::get_class ( ) [inline]
```

Definition at line 161 of file Module\_Property.h.

References CppProperty\_GetMethod\_SetPointer< Class, PROP >::class\_name.

### 6.129.4.3 is\_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetMethod_SetPointer< Class, PROP >::is_readonly ( ) [inline]
```

Definition at line 160 of file Module\_Property.h.

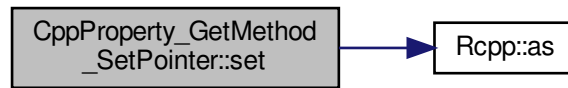
#### 6.129.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetMethod_SetPointer< Class, PROP >::set (
    Class * object,
    SEXP value ) [inline]
```

Definition at line 157 of file Module\_Property.h.

References Rcpp::as(), and CppProperty\_GetMethod\_SetPointer< Class, PROP >::setter.

Here is the call graph for this function:



### 6.129.5 Member Data Documentation

#### 6.129.5.1 class\_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetMethod_SetPointer< Class, PROP >::class_name [private]
```

Definition at line 166 of file Module\_Property.h.

Referenced by CppProperty\_GetMethod\_SetPointer< Class, PROP >::get\_class().

#### 6.129.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetMethod_SetPointer< Class, PROP >::getter [private]
```

Definition at line 164 of file Module\_Property.h.

Referenced by CppProperty\_GetMethod\_SetPointer< Class, PROP >::get().

### 6.129.5.3 setter

```
template<typename Class , typename PROP >
SetMethod CppProperty_GetMethod_SetPointer< Class, PROP >::setter [private]
```

Definition at line 165 of file Module\_Property.h.

Referenced by CppProperty\_GetMethod\_SetPointer< Class, PROP >::set().

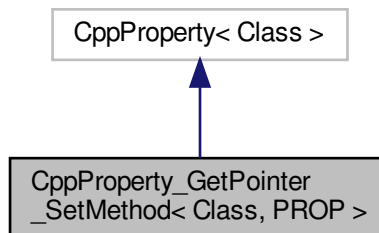
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module\_Property.h

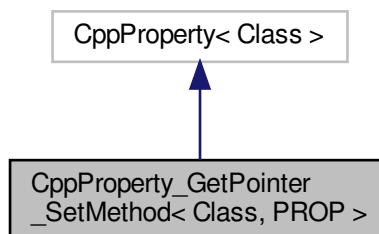
## 6.130 CppProperty\_GetPointer\_SetMethod< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty\_GetPointer\_SetMethod< Class, PROP >:



Collaboration diagram for CppProperty\_GetPointer\_SetMethod< Class, PROP >:



## Public Types

- typedef PROP(\* [GetMethod](#)) (Class \*)
- typedef void(Class::\* [SetMethod](#)) (PROP)
- typedef CppProperty< Class > [prop\\_class](#)

## Public Member Functions

- [CppProperty\\_GetPointer\\_SetMethod](#) ([GetMethod](#) getter\_, [SetMethod](#) setter\_, const char \*doc=0)
- SEXP [get](#) (Class \*object)
- void [set](#) (Class \*object, SEXP value)
- bool [is\\_readonly](#) ()
- std::string [get\\_class](#) ()

## Private Attributes

- [GetMethod](#) [getter](#)
- [SetMethod](#) [setter](#)
- std::string [class\\_name](#)

### 6.130.1 Detailed Description

```
template<typename Class, typename PROP>
class CppProperty_GetPointer_SetMethod< Class, PROP >
```

Definition at line 197 of file Module\_Property.h.

### 6.130.2 Member Typedef Documentation

#### 6.130.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(* CppProperty\_GetPointer\_SetMethod< Class, PROP >::GetMethod) (Class *)
```

Definition at line 199 of file Module\_Property.h.

### 6.130.2.2 prop\_class

```
template<typename Class , typename PROP >  
typedef CppProperty<Class> CppProperty_GetPointer_SetMethod< Class, PROP >::prop_class
```

Definition at line 201 of file Module\_Property.h.

### 6.130.2.3 SetMethod

```
template<typename Class , typename PROP >  
typedef void(Class::* CppProperty_GetPointer_SetMethod< Class, PROP >::SetMethod) (PROP)
```

Definition at line 200 of file Module\_Property.h.

## 6.130.3 Constructor & Destructor Documentation

### 6.130.3.1 CppProperty\_GetPointer\_SetMethod()

```
template<typename Class , typename PROP >  
CppProperty_GetPointer_SetMethod< Class, PROP >::CppProperty_GetPointer_SetMethod (  
    GetMethod getter_,  
    SetMethod setter_,  
    const char * doc = 0 ) [inline]
```

Definition at line 203 of file Module\_Property.h.

## 6.130.4 Member Function Documentation



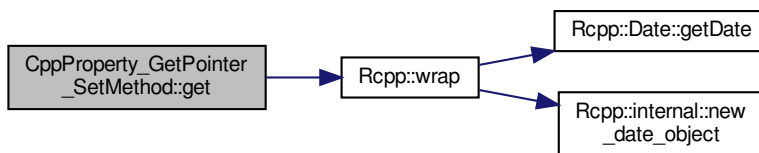
#### 6.130.4.1 get()

```
template<typename Class , typename PROP >
SEXPR CppProperty_GetPointer_SetMethod< Class, PROP >::get (
    Class * object ) [inline]
```

Definition at line 206 of file Module\_Property.h.

References CppProperty\_GetPointer\_SetMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



#### 6.130.4.2 get\_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetPointer_SetMethod< Class, PROP >::get_class ( ) [inline]
```

Definition at line 213 of file Module\_Property.h.

References CppProperty\_GetPointer\_SetMethod< Class, PROP >::class\_name.

#### 6.130.4.3 is\_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetPointer_SetMethod< Class, PROP >::is_readonly ( ) [inline]
```

Definition at line 212 of file Module\_Property.h.

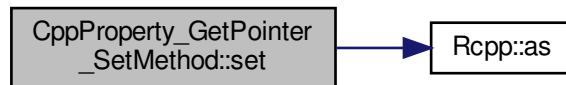
#### 6.130.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetPointer_SetMethod< Class, PROP >::set (
    Class * object,
    SEXP value ) [inline]
```

Definition at line 209 of file Module\_Property.h.

References Rcpp::as(), and CppProperty\_GetPointer\_SetMethod< Class, PROP >::setter.

Here is the call graph for this function:



### 6.130.5 Member Data Documentation

#### 6.130.5.1 class\_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetPointer_SetMethod< Class, PROP >::class_name [private]
```

Definition at line 218 of file Module\_Property.h.

Referenced by CppProperty\_GetPointer\_SetMethod< Class, PROP >::get\_class().

#### 6.130.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetPointer_SetMethod< Class, PROP >::getter [private]
```

Definition at line 216 of file Module\_Property.h.

Referenced by CppProperty\_GetPointer\_SetMethod< Class, PROP >::get().

## 6.130.5.3 setter

```
template<typename Class , typename PROP >
SetMethod CppProperty_GetPointer_SetMethod< Class, PROP >::setter [private]
```

Definition at line 217 of file Module\_Property.h.

Referenced by CppProperty\_GetPointer\_SetMethod< Class, PROP >::set().

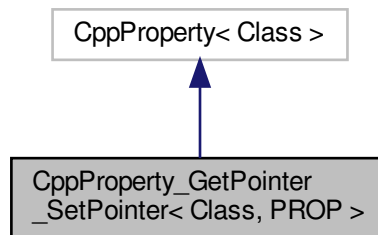
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/module/Module\\_Property.h](#)

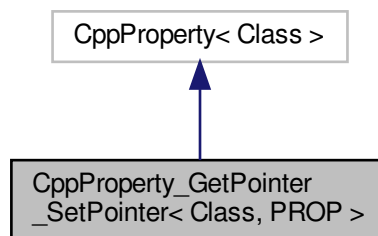
## 6.131 CppProperty\_GetPointer\_SetPointer< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty\_GetPointer\_SetPointer< Class, PROP >:



Collaboration diagram for CppProperty\_GetPointer\_SetPointer< Class, PROP >:



## Public Types

- typedef PROP(\* [GetMethod](#)) (Class \*)
- typedef void(\* [SetMethod](#)) (Class \*, PROP)
- typedef CppProperty< Class > [prop\\_class](#)

## Public Member Functions

- [CppProperty\\_GetPointer\\_SetPointer](#) ([GetMethod](#) getter\_, [SetMethod](#) setter\_, const char \*doc=0)
- SEXP [get](#) (Class \*object)
- void [set](#) (Class \*object, SEXP value)
- bool [is\\_readonly](#) ()
- std::string [get\\_class](#) ()

## Private Attributes

- [GetMethod](#) [getter](#)
- [SetMethod](#) [setter](#)
- std::string [class\\_name](#)

### 6.131.1 Detailed Description

```
template<typename Class, typename PROP>
class CppProperty_GetPointer_SetPointer< Class, PROP >
```

Definition at line 225 of file Module\_Property.h.

### 6.131.2 Member Typedef Documentation

#### 6.131.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(* CppProperty\_GetPointer\_SetPointer< Class, PROP >::GetMethod) (Class *)
```

Definition at line 227 of file Module\_Property.h.

### 6.131.2.2 prop\_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty_GetPointer_SetPointer< Class, PROP >::prop_class
```

Definition at line 229 of file Module\_Property.h.

### 6.131.2.3 SetMethod

```
template<typename Class , typename PROP >
typedef void(* CppProperty_GetPointer_SetPointer< Class, PROP >::SetMethod) (Class *, PROP)
```

Definition at line 228 of file Module\_Property.h.

## 6.131.3 Constructor & Destructor Documentation

### 6.131.3.1 CppProperty\_GetPointer\_SetPointer()

```
template<typename Class , typename PROP >
CppProperty_GetPointer_SetPointer< Class, PROP >::CppProperty_GetPointer_SetPointer (
    GetMethod getter_,
    SetMethod setter_,
    const char * doc = 0 ) [inline]
```

Definition at line 231 of file Module\_Property.h.

### 6.131.4 Member Function Documentation

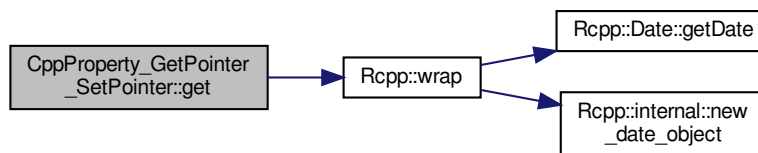
### 6.131.4.1 get()

```
template<typename Class , typename PROP >
SEXPR CppProperty_GetPointer_SetPointer< Class, PROP >::get (
    Class * object ) [inline]
```

Definition at line 234 of file Module\_Property.h.

References CppProperty\_GetPointer\_SetPointer< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



### 6.131.4.2 get\_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetPointer_SetPointer< Class, PROP >::get_class ( ) [inline]
```

Definition at line 241 of file Module\_Property.h.

References CppProperty\_GetPointer\_SetPointer< Class, PROP >::class\_name.

### 6.131.4.3 is\_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetPointer_SetPointer< Class, PROP >::is_readonly ( ) [inline]
```

Definition at line 240 of file Module\_Property.h.

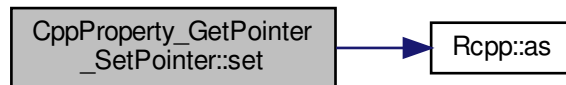
#### 6.131.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetPointer_SetPointer< Class, PROP >::set (
    Class * object,
    SEXP value ) [inline]
```

Definition at line 237 of file Module\_Property.h.

References Rcpp::as(), and CppProperty\_GetPointer\_SetPointer< Class, PROP >::setter.

Here is the call graph for this function:



### 6.131.5 Member Data Documentation

#### 6.131.5.1 class\_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetPointer_SetPointer< Class, PROP >::class_name [private]
```

Definition at line 246 of file Module\_Property.h.

Referenced by CppProperty\_GetPointer\_SetPointer< Class, PROP >::get\_class().

#### 6.131.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetPointer_SetPointer< Class, PROP >::getter [private]
```

Definition at line 244 of file Module\_Property.h.

Referenced by CppProperty\_GetPointer\_SetPointer< Class, PROP >::get().

### 6.131.5.3 setter

```
template<typename Class , typename PROP >
SetMethod CppProperty_GetPointer_SetPointer< Class, PROP >::setter [private]
```

Definition at line 245 of file Module\_Property.h.

Referenced by CppProperty\_GetPointer\_SetPointer< Class, PROP >::set().

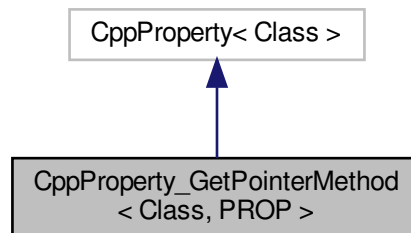
The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module\_Property.h

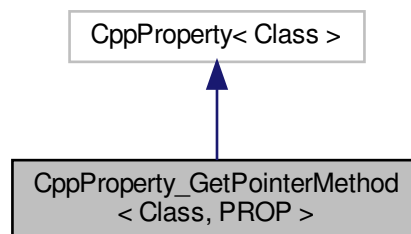
## 6.132 CppProperty\_GetPointerMethod< Class, PROP > Class Template Reference

```
#include <Module_Property.h>
```

Inheritance diagram for CppProperty\_GetPointerMethod< Class, PROP >:



Collaboration diagram for CppProperty\_GetPointerMethod< Class, PROP >:





## Public Types

- typedef PROP(\* [GetMethod](#)) (Class \*)
- typedef CppProperty< Class > [prop\\_class](#)

## Public Member Functions

- [CppProperty\\_GetPointerMethod](#) ([GetMethod](#) getter\_, const char \*doc=0)
- SEXP [get](#) (Class \*object)
- void [set](#) (Class \*, SEXP)
- bool [is\\_readonly](#) ()
- std::string [get\\_class](#) ()

## Private Attributes

- [GetMethod](#) getter
- std::string [class\\_name](#)

### 6.132.1 Detailed Description

```
template<typename Class, typename PROP>
class CppProperty_GetPointerMethod< Class, PROP >
```

Definition at line 70 of file Module\_Property.h.

### 6.132.2 Member Typedef Documentation

#### 6.132.2.1 GetMethod

```
template<typename Class , typename PROP >
typedef PROP(* CppProperty\_GetPointerMethod< Class, PROP >::GetMethod) (Class *)
```

Definition at line 72 of file Module\_Property.h.

#### 6.132.2.2 prop\_class

```
template<typename Class , typename PROP >
typedef CppProperty<Class> CppProperty\_GetPointerMethod< Class, PROP >::prop_class
```

Definition at line 73 of file Module\_Property.h.

### 6.132.3 Constructor & Destructor Documentation

#### 6.132.3.1 CppProperty\_GetPointerMethod()

```
template<typename Class , typename PROP >
CppProperty_GetPointerMethod< Class, PROP >::CppProperty_GetPointerMethod (
    GetMethod getter_,
    const char * doc = 0 ) [inline]
```

Definition at line 75 of file Module\_Property.h.

### 6.132.4 Member Function Documentation

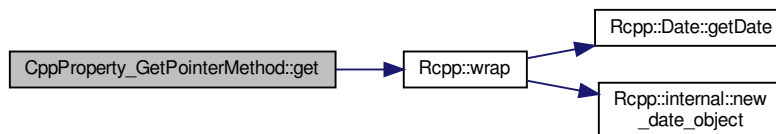
#### 6.132.4.1 get()

```
template<typename Class , typename PROP >
SEXP CppProperty_GetPointerMethod< Class, PROP >::get (
    Class * object ) [inline]
```

Definition at line 78 of file Module\_Property.h.

References CppProperty\_GetPointerMethod< Class, PROP >::getter, and Rcpp::wrap().

Here is the call graph for this function:



#### 6.132.4.2 get\_class()

```
template<typename Class , typename PROP >
std::string CppProperty_GetPointerMethod< Class, PROP >::get_class ( ) [inline]
```

Definition at line 81 of file Module\_Property.h.

References CppProperty\_GetPointerMethod< Class, PROP >::class\_name.

#### 6.132.4.3 is\_readonly()

```
template<typename Class , typename PROP >
bool CppProperty_GetPointerMethod< Class, PROP >::is_readonly ( ) [inline]
```

Definition at line 80 of file Module\_Property.h.

#### 6.132.4.4 set()

```
template<typename Class , typename PROP >
void CppProperty_GetPointerMethod< Class, PROP >::set (
    Class * ,
    SEXP ) [inline]
```

Definition at line 79 of file Module\_Property.h.

### 6.132.5 Member Data Documentation

#### 6.132.5.1 class\_name

```
template<typename Class , typename PROP >
std::string CppProperty_GetPointerMethod< Class, PROP >::class_name [private]
```

Definition at line 85 of file Module\_Property.h.

Referenced by CppProperty\_GetPointerMethod< Class, PROP >::get\_class().

### 6.132.5.2 getter

```
template<typename Class , typename PROP >
GetMethod CppProperty_GetPointerMethod< Class, PROP >::getter [private]
```

Definition at line 84 of file Module\_Property.h.

Referenced by CppProperty\_GetPointerMethod< Class, PROP >::get().

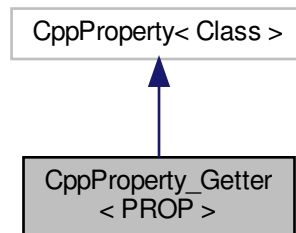
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/module/Module\\_Property.h](#)

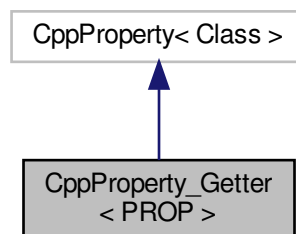
## 6.133 CppProperty\_Getter< PROP > Class Template Reference

```
#include <Module_Field.h>
```

Inheritance diagram for CppProperty\_Getter< PROP >:



Collaboration diagram for CppProperty\_Getter< PROP >:



## Public Types

- typedef PROP Class::\* [pointer](#)
- typedef CppProperty< Class > [prop\\_class](#)

## Public Member Functions

- [CppProperty\\_Getter](#) ([pointer](#) ptr\_, const char \*doc=0)
- SEXP [get](#) (Class \*object)
- void [set](#) (Class \*object, SEXP value)
- bool [is\\_readonly](#) ()
- std::string [get\\_class](#) ()

## Private Attributes

- [pointer](#) ptr
- std::string [class\\_name](#)

### 6.133.1 Detailed Description

```
template<typename PROP>  
class CppProperty_Getter< PROP >
```

Definition at line 48 of file Module\_Field.h.

### 6.133.2 Member Typedef Documentation

#### 6.133.2.1 pointer

```
template<typename PROP >  
typedef PROP Class::* CppProperty\_Getter< PROP >::pointer
```

Definition at line 50 of file Module\_Field.h.

#### 6.133.2.2 prop\_class

```
template<typename PROP >  
typedef CppProperty<Class> CppProperty\_Getter< PROP >::prop\_class
```

Definition at line 51 of file Module\_Field.h.

### 6.133.3 Constructor & Destructor Documentation

#### 6.133.3.1 CppProperty\_Getter()

```
template<typename PROP >
CppProperty_Getter< PROP >::CppProperty_Getter (
    pointer ptr_,
    const char * doc = 0 ) [inline]
```

Definition at line 53 of file Module\_Field.h.

### 6.133.4 Member Function Documentation

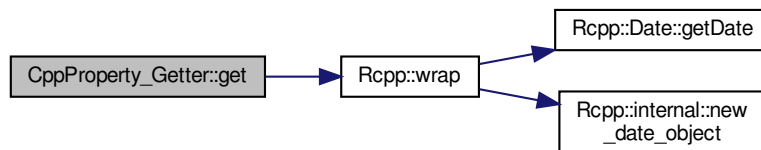
#### 6.133.4.1 get()

```
template<typename PROP >
SEXP CppProperty_Getter< PROP >::get (
    Class * object ) [inline]
```

Definition at line 56 of file Module\_Field.h.

References CppProperty\_Getter< PROP >::ptr, and Rcpp::wrap().

Here is the call graph for this function:



#### 6.133.4.2 get\_class()

```
template<typename PROP >
std::string CppProperty_Getter< PROP >::get_class ( ) [inline]
```

Definition at line 59 of file Module\_Field.h.

References CppProperty\_Getter< PROP >::class\_name.

#### 6.133.4.3 is\_readonly()

```
template<typename PROP >
bool CppProperty_Getter< PROP >::is_readonly ( ) [inline]
```

Definition at line 58 of file Module\_Field.h.

#### 6.133.4.4 set()

```
template<typename PROP >
void CppProperty_Getter< PROP >::set (
    Class * object,
    SEXP value ) [inline]
```

Definition at line 57 of file Module\_Field.h.

### 6.133.5 Member Data Documentation

#### 6.133.5.1 class\_name

```
template<typename PROP >
std::string CppProperty_Getter< PROP >::class_name [private]
```

Definition at line 63 of file Module\_Field.h.

Referenced by CppProperty\_Getter< PROP >::get\_class().

### 6.133.5.2 ptr

```
template<typename PROP >  
pointer CppProperty_Getter< PROP >::ptr [private]
```

Definition at line 62 of file Module\_Field.h.

Referenced by CppProperty\_Getter< PROP >::get().

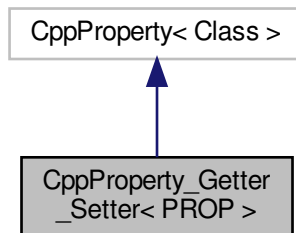
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/module/Module\\_Field.h](#)

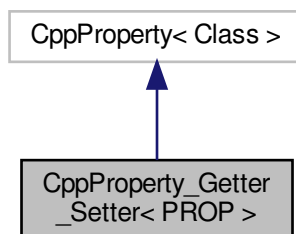
## 6.134 CppProperty\_Getter\_Setter< PROP > Class Template Reference

```
#include <Module_Field.h>
```

Inheritance diagram for CppProperty\_Getter\_Setter< PROP >:



Collaboration diagram for CppProperty\_Getter\_Setter< PROP >:





## Public Types

- typedef PROP Class::\* [pointer](#)
- typedef CppProperty< Class > [prop\\_class](#)

## Public Member Functions

- [CppProperty\\_Getter\\_Setter](#) ([pointer](#) ptr\_, const char \*doc)
- SEXP [get](#) (Class \*object)
- void [set](#) (Class \*object, SEXP value)
- bool [is\\_readonly](#) ()
- std::string [get\\_class](#) ()

## Private Attributes

- [pointer](#) ptr
- std::string [class\\_name](#)

### 6.134.1 Detailed Description

```
template<typename PROP>
class CppProperty_Getter_Setter< PROP >
```

Definition at line 27 of file Module\_Field.h.

### 6.134.2 Member Typedef Documentation

#### 6.134.2.1 pointer

```
template<typename PROP >
typedef PROP Class::* CppProperty\_Getter\_Setter< PROP >::pointer
```

Definition at line 29 of file Module\_Field.h.

#### 6.134.2.2 prop\_class

```
template<typename PROP >
typedef CppProperty<Class> CppProperty\_Getter\_Setter< PROP >::prop\_class
```

Definition at line 30 of file Module\_Field.h.

### 6.134.3 Constructor & Destructor Documentation

#### 6.134.3.1 CppProperty\_Getter\_Setter()

```
template<typename PROP >
CppProperty_Getter_Setter< PROP >::CppProperty_Getter_Setter (
    pointer ptr_,
    const char * doc ) [inline]
```

Definition at line 32 of file Module\_Field.h.

### 6.134.4 Member Function Documentation

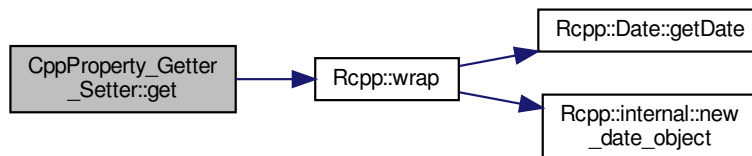
#### 6.134.4.1 get()

```
template<typename PROP >
SEXP CppProperty_Getter_Setter< PROP >::get (
    Class * object ) [inline]
```

Definition at line 35 of file Module\_Field.h.

References CppProperty\_Getter\_Setter< PROP >::ptr, and Rcpp::wrap().

Here is the call graph for this function:



#### 6.134.4.2 get\_class()

```
template<typename PROP >
std::string CppProperty_Getter_Setter< PROP >::get_class ( ) [inline]
```

Definition at line 38 of file Module\_Field.h.

References CppProperty\_Getter\_Setter< PROP >::class\_name.

#### 6.134.4.3 is\_readonly()

```
template<typename PROP >
bool CppProperty_Getter_Setter< PROP >::is_readonly ( ) [inline]
```

Definition at line 37 of file Module\_Field.h.

#### 6.134.4.4 set()

```
template<typename PROP >
void CppProperty_Getter_Setter< PROP >::set (
    Class * object,
    SEXP value ) [inline]
```

Definition at line 36 of file Module\_Field.h.

References CppProperty\_Getter\_Setter< PROP >::ptr.

### 6.134.5 Member Data Documentation

#### 6.134.5.1 class\_name

```
template<typename PROP >
std::string CppProperty_Getter_Setter< PROP >::class_name [private]
```

Definition at line 42 of file Module\_Field.h.

Referenced by CppProperty\_Getter\_Setter< PROP >::get\_class().

### 6.134.5.2 ptr

```
template<typename PROP >
pointer CppProperty_Getter_Setter< PROP >::ptr [private]
```

Definition at line 41 of file Module\_Field.h.

Referenced by CppProperty\_Getter\_Setter< PROP >::get(), and CppProperty\_Getter\_Setter< PROP >::set().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/module/Module\_Field.h

## 6.135 Rcpp::algorithm::helpers::ctype< T > Struct Template Reference

```
#include <algorithm.h>
```

### Public Types

- typedef `ctype_helper< sizeof(test(make())) >::type` type

### Static Public Member Functions

- static `CTYPE_CHAR test` (const char &)
- static `CTYPE_SHORT test` (const short &)
- static `CTYPE_INT test` (const int &)
- static `CTYPE_LONG test` (const long &)
- static `CTYPE_FLOAT test` (const float &)
- static `CTYPE_DOUBLE test` (const double &)
- static `CTYPE_LONG_DOUBLE test` (const long double &)
- static `CTYPE_STRING test` (const std::string &)
- static `CTYPE_UNSIGNED_CHAR test` (const unsigned char &)
- static `CTYPE_UNSIGNED_SHORT test` (const unsigned short &)
- static `CTYPE_UNSIGNED_INT test` (const unsigned int &)
- static `CTYPE_UNSIGNED_LONG test` (const unsigned long &)
- static `CTYPE_UNKNOWN test` (...)
- static `T make` ()

### 6.135.1 Detailed Description

```
template<typename T>
struct Rcpp::algorithm::helpers::ctype< T >
```

Definition at line 108 of file algorithm.h.

## 6.135.2 Member Typedef Documentation

### 6.135.2.1 type

```
template<typename T >
typedef ctype\_helper< sizeof(test(make())) >::type Rcpp::algorithm::helpers::ctype< T >::type
```

Definition at line 132 of file `algorithm.h`.

## 6.135.3 Member Function Documentation

### 6.135.3.1 make()

```
template<typename T >
static T Rcpp::algorithm::helpers::ctype< T >::make ( ) [static]
```

### 6.135.3.2 test() [1/13]

```
template<typename T >
static CTYPE\_UNKNOWN Rcpp::algorithm::helpers::ctype< T >::test (
    ... ) [static]
```

### 6.135.3.3 test() [2/13]

```
template<typename T >
static CTYPE\_CHAR Rcpp::algorithm::helpers::ctype< T >::test (
    const char & ) [static]
```

### 6.135.3.4 test() [3/13]

```
template<typename T >
static CTYPE\_DOUBLE Rcpp::algorithm::helpers::ctype< T >::test (
    const double & ) [static]
```

**6.135.3.5 test()** [4/13]

```
template<typename T >
static CTYPE_FLOAT Rcpp::algorithm::helpers::ctype< T >::test (
    const float & ) [static]
```

**6.135.3.6 test()** [5/13]

```
template<typename T >
static CTYPE_INT Rcpp::algorithm::helpers::ctype< T >::test (
    const int & ) [static]
```

**6.135.3.7 test()** [6/13]

```
template<typename T >
static CTYPE_LONG Rcpp::algorithm::helpers::ctype< T >::test (
    const long & ) [static]
```

**6.135.3.8 test()** [7/13]

```
template<typename T >
static CTYPE_LONG_DOUBLE Rcpp::algorithm::helpers::ctype< T >::test (
    const long double & ) [static]
```

**6.135.3.9 test()** [8/13]

```
template<typename T >
static CTYPE_SHORT Rcpp::algorithm::helpers::ctype< T >::test (
    const short & ) [static]
```

**6.135.3.10 test()** [9/13]

```
template<typename T >
static CTYPE_STRING Rcpp::algorithm::helpers::ctype< T >::test (
    const std::string & ) [static]
```

**6.135.3.11 test()** [10/13]

```
template<typename T >
static CTYPE_UNSIGNED_CHAR Rcpp::algorithm::helpers::ctype< T >::test (
    const unsigned char & ) [static]
```

**6.135.3.12 test()** [11/13]

```
template<typename T >
static CTYPE_UNSIGNED_INT Rcpp::algorithm::helpers::ctype< T >::test (
    const unsigned int & ) [static]
```

**6.135.3.13 test()** [12/13]

```
template<typename T >
static CTYPE_UNSIGNED_LONG Rcpp::algorithm::helpers::ctype< T >::test (
    const unsigned long & ) [static]
```

**6.135.3.14 test()** [13/13]

```
template<typename T >
static CTYPE_UNSIGNED_SHORT Rcpp::algorithm::helpers::ctype< T >::test (
    const unsigned short & ) [static]
```

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

**6.136 Rcpp::algorithm::helpers::CTYPE\_CHAR Struct Reference**

```
#include <algorithm.h>
```

**Public Attributes**

- char [a](#) [1]

### 6.136.1 Detailed Description

Definition at line 37 of file algorithm.h.

### 6.136.2 Member Data Documentation

#### 6.136.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_CHAR::a[1]
```

Definition at line 37 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.137 Rcpp::algorithm::helpers::CTYPE\_DOUBLE Struct Reference

```
#include <algorithm.h>
```

### Public Attributes

- char [a](#) [7]

### 6.137.1 Detailed Description

Definition at line 45 of file algorithm.h.

### 6.137.2 Member Data Documentation

#### 6.137.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_DOUBLE::a[7]
```

Definition at line 45 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)



## 6.138 Rcpp::algorithm::helpers::CTYPE\_FLOAT Struct Reference

```
#include <algorithm.h>
```

### Public Attributes

- char [a](#) [6]

### 6.138.1 Detailed Description

Definition at line 44 of file algorithm.h.

### 6.138.2 Member Data Documentation

#### 6.138.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_FLOAT::a[6]
```

Definition at line 44 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.139 Rcpp::algorithm::helpers::ctype\_helper< I > Struct Template Reference

```
#include <algorithm.h>
```

### Static Public Attributes

- static const bool [value](#) = false

### 6.139.1 Detailed Description

```
template<std::size_t I>  
struct Rcpp::algorithm::helpers::ctype_helper< I >
```

Definition at line 58 of file algorithm.h.

## 6.139.2 Member Data Documentation

### 6.139.2.1 value

```
template<std::size_t I>
const bool Rcpp::algorithm::helpers::ctype_helper< I >::value = false [static]
```

Definition at line 58 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.140 Rcpp::algorithm::helpers::ctype\_helper< sizeof(CTYPE\_CHAR) > Struct Reference

```
#include <algorithm.h>
```

### Public Types

- typedef char [type](#)

### Static Public Attributes

- static const bool [value](#) = true

### 6.140.1 Detailed Description

Definition at line 61 of file algorithm.h.

## 6.140.2 Member Typedef Documentation

### 6.140.2.1 type

```
typedef char Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_CHAR) >::type
```

Definition at line 61 of file algorithm.h.

## 6.140.3 Member Data Documentation

### 6.140.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_CHAR) >::value = true [static]
```

Definition at line 61 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.141 Rcpp::algorithm::helpers::ctype\_helper< sizeof(CTYPE\_DOUBLE) > Struct Reference

```
#include <algorithm.h>
```

### Public Types

- typedef double [type](#)

### Static Public Attributes

- static const bool [value](#) = true

### 6.141.1 Detailed Description

Definition at line 81 of file algorithm.h.

## 6.141.2 Member Typedef Documentation

### 6.141.2.1 type

```
typedef double Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_DOUBLE) >::type
```

Definition at line 81 of file algorithm.h.

### 6.141.3 Member Data Documentation

#### 6.141.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_DOUBLE) >::value = true [static]
```

Definition at line 81 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

### 6.142 Rcpp::algorithm::helpers::ctype\_helper< sizeof(CTYPE\_FLOAT) > Struct Reference

```
#include <algorithm.h>
```

#### Public Types

- typedef float [type](#)

#### Static Public Attributes

- static const bool [value](#) = true

#### 6.142.1 Detailed Description

Definition at line 78 of file algorithm.h.

#### 6.142.2 Member Typedef Documentation

##### 6.142.2.1 type

```
typedef float Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_FLOAT) >::type
```

Definition at line 78 of file algorithm.h.

## 6.142.3 Member Data Documentation

### 6.142.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_FLOAT) >::value = true [static]
```

Definition at line 78 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.143 Rcpp::algorithm::helpers::ctype\_helper< sizeof(CTYPE\_INT) > Struct Reference

```
#include <algorithm.h>
```

### Public Types

- typedef int [type](#)

### Static Public Attributes

- static const bool [value](#) = true

### 6.143.1 Detailed Description

Definition at line 67 of file algorithm.h.

## 6.143.2 Member Typedef Documentation

### 6.143.2.1 type

```
typedef int Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_INT) >::type
```

Definition at line 67 of file algorithm.h.

### 6.143.3 Member Data Documentation

#### 6.143.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_INT) >::value = true [static]
```

Definition at line 67 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

### 6.144 Rcpp::algorithm::helpers::ctype\_helper< sizeof(CTYPE\_LONG) > Struct Reference

```
#include <algorithm.h>
```

#### Public Types

- typedef long [type](#)

#### Static Public Attributes

- static const bool [value](#) = true

#### 6.144.1 Detailed Description

Definition at line 70 of file algorithm.h.

#### 6.144.2 Member Typedef Documentation

##### 6.144.2.1 type

```
typedef long Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG) >::type
```

Definition at line 70 of file algorithm.h.

## 6.144.3 Member Data Documentation

### 6.144.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG) >::value = true [static]
```

Definition at line 70 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.145 Rcpp::algorithm::helpers::ctype\_helper< sizeof(CTYPE\_LONG\_DOUBLE) > Struct Reference

```
#include <algorithm.h>
```

### Public Types

- typedef long double [type](#)

### Static Public Attributes

- static const bool [value](#) = true

### 6.145.1 Detailed Description

Definition at line 84 of file algorithm.h.

## 6.145.2 Member Typedef Documentation

### 6.145.2.1 type

```
typedef long double Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG_DOUBLE) >::type
```

Definition at line 84 of file algorithm.h.

### 6.145.3 Member Data Documentation

#### 6.145.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_LONG_DOUBLE) >::value = true
[static]
```

Definition at line 84 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

### 6.146 Rcpp::algorithm::helpers::ctype\_helper< sizeof(CTYPE\_SHORT) > Struct Reference

```
#include <algorithm.h>
```

#### Public Types

- typedef short [type](#)

#### Static Public Attributes

- static const bool [value](#) = true

#### 6.146.1 Detailed Description

Definition at line 64 of file algorithm.h.

### 6.146.2 Member Typedef Documentation

#### 6.146.2.1 type

```
typedef short Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_SHORT) >::type
```

Definition at line 64 of file algorithm.h.



## 6.146.3 Member Data Documentation

### 6.146.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_SHORT) >::value = true [static]
```

Definition at line 64 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.147 Rcpp::algorithm::helpers::ctype\_helper< sizeof(CTYPE\_STRING) > Struct Reference

```
#include <algorithm.h>
```

### Public Types

- typedef std::string [type](#)

### Static Public Attributes

- static const bool [value](#) = true

### 6.147.1 Detailed Description

Definition at line 87 of file algorithm.h.

### 6.147.2 Member Typedef Documentation

#### 6.147.2.1 type

```
typedef std::string Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_STRING) >::type
```

Definition at line 87 of file algorithm.h.

### 6.147.3 Member Data Documentation

#### 6.147.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_STRING) >::value = true [static]
```

Definition at line 87 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

### 6.148 Rcpp::algorithm::helpers::ctype\_helper< sizeof(CTYPE\_UNSIGNED\_CHAR) > Struct Reference

```
#include <algorithm.h>
```

#### Public Types

- typedef unsigned char [type](#)

#### Static Public Attributes

- static const bool [value](#) = true

#### 6.148.1 Detailed Description

Definition at line 90 of file algorithm.h.

#### 6.148.2 Member Typedef Documentation

##### 6.148.2.1 type

```
typedef unsigned char Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) >::type
```

Definition at line 90 of file algorithm.h.

### 6.148.3 Member Data Documentation

#### 6.148.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_CHAR) >::value = true  
[static]
```

Definition at line 90 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.149 Rcpp::algorithm::helpers::ctype\_helper< sizeof(CTYPE\_UNSIGNED\_INT) > Struct Reference

```
#include <algorithm.h>
```

### Public Types

- typedef unsigned int [type](#)

### Static Public Attributes

- static const bool [value](#) = true

### 6.149.1 Detailed Description

Definition at line 96 of file algorithm.h.

### 6.149.2 Member Typedef Documentation

#### 6.149.2.1 type

```
typedef unsigned int Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_INT) >::type
```

Definition at line 96 of file algorithm.h.

### 6.149.3 Member Data Documentation

#### 6.149.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_INT) >::value = true
[static]
```

Definition at line 96 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

### 6.150 Rcpp::algorithm::helpers::ctype\_helper< sizeof(CTYPE\_UNSIGNED\_LONG) > Struct Reference

```
#include <algorithm.h>
```

#### Public Types

- typedef unsigned long [type](#)

#### Static Public Attributes

- static const bool [value](#) = true

#### 6.150.1 Detailed Description

Definition at line 99 of file algorithm.h.

### 6.150.2 Member Typedef Documentation

#### 6.150.2.1 type

```
typedef unsigned long Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) >::type
```

Definition at line 99 of file algorithm.h.

### 6.150.3 Member Data Documentation

#### 6.150.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_LONG) >::value = true
[static]
```

Definition at line 99 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.151 Rcpp::algorithm::helpers::ctype\_helper< sizeof(CTYPE\_UNSIGNED\_SHORT) > Struct Reference

```
#include <algorithm.h>
```

### Public Types

- typedef unsigned short [type](#)

### Static Public Attributes

- static const bool [value](#) = true

#### 6.151.1 Detailed Description

Definition at line 93 of file algorithm.h.

#### 6.151.2 Member Typedef Documentation

##### 6.151.2.1 type

```
typedef unsigned short Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) >↔
::type
```

Definition at line 93 of file algorithm.h.

### 6.151.3 Member Data Documentation

#### 6.151.3.1 value

```
const bool Rcpp::algorithm::helpers::ctype_helper< sizeof(CTYPE_UNSIGNED_SHORT) >::value = true
[static]
```

Definition at line 93 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

### 6.152 Rcpp::algorithm::helpers::CTYPE\_INT Struct Reference

```
#include <algorithm.h>
```

#### Public Attributes

- char [a](#) [3]

#### 6.152.1 Detailed Description

Definition at line 39 of file algorithm.h.

### 6.152.2 Member Data Documentation

#### 6.152.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_INT::a[3]
```

Definition at line 39 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.153 Rcpp::algorithm::helpers::CTYPE\_LONG Struct Reference

```
#include <algorithm.h>
```

### Public Attributes

- char [a](#) [4]

#### 6.153.1 Detailed Description

Definition at line 40 of file `algorithm.h`.

#### 6.153.2 Member Data Documentation

##### 6.153.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_LONG::a[4]
```

Definition at line 40 of file `algorithm.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.154 Rcpp::algorithm::helpers::CTYPE\_LONG\_DOUBLE Struct Reference

```
#include <algorithm.h>
```

### Public Attributes

- char [a](#) [8]

#### 6.154.1 Detailed Description

Definition at line 46 of file `algorithm.h`.

## 6.154.2 Member Data Documentation

### 6.154.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_LONG_DOUBLE::a[8]
```

Definition at line 46 of file `algorithm.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.155 Rcpp::algorithm::helpers::CTYPE\_SHORT Struct Reference

```
#include <algorithm.h>
```

### Public Attributes

- char `a` [2]

### 6.155.1 Detailed Description

Definition at line 38 of file `algorithm.h`.

## 6.155.2 Member Data Documentation

### 6.155.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_SHORT::a[2]
```

Definition at line 38 of file `algorithm.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)



## 6.156 Rcpp::algorithm::helpers::CTYPE\_STRING Struct Reference

```
#include <algorithm.h>
```

### Public Attributes

- char [a](#) [9]

#### 6.156.1 Detailed Description

Definition at line 47 of file `algorithm.h`.

#### 6.156.2 Member Data Documentation

##### 6.156.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_STRING::a[9]
```

Definition at line 47 of file `algorithm.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.157 Rcpp::algorithm::helpers::CTYPE\_UNKNOWN Struct Reference

```
#include <algorithm.h>
```

### Public Attributes

- char [a](#) [128]

#### 6.157.1 Detailed Description

Definition at line 55 of file `algorithm.h`.

## 6.157.2 Member Data Documentation

### 6.157.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_UNKNOWN::a[128]
```

Definition at line 55 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.158 Rcpp::algorithm::helpers::CTYPE\_UNSIGNED\_CHAR Struct Reference

```
#include <algorithm.h>
```

### Public Attributes

- char [a](#) [10]

### 6.158.1 Detailed Description

Definition at line 48 of file algorithm.h.

## 6.158.2 Member Data Documentation

### 6.158.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_UNSIGNED_CHAR::a[10]
```

Definition at line 48 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.159 Rcpp::algorithm::helpers::CTYPE\_UNSIGNED\_INT Struct Reference

```
#include <algorithm.h>
```

### Public Attributes

- char [a](#) [12]

#### 6.159.1 Detailed Description

Definition at line 50 of file `algorithm.h`.

#### 6.159.2 Member Data Documentation

##### 6.159.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_UNSIGNED_INT::a[12]
```

Definition at line 50 of file `algorithm.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.160 Rcpp::algorithm::helpers::CTYPE\_UNSIGNED\_LONG Struct Reference

```
#include <algorithm.h>
```

### Public Attributes

- char [a](#) [13]

#### 6.160.1 Detailed Description

Definition at line 51 of file `algorithm.h`.

## 6.160.2 Member Data Documentation

### 6.160.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_UNSIGNED_LONG::a[13]
```

Definition at line 51 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.161 Rcpp::algorithm::helpers::CTYPE\_UNSIGNED\_SHORT Struct Reference

```
#include <algorithm.h>
```

### Public Attributes

- char [a](#) [11]

### 6.161.1 Detailed Description

Definition at line 49 of file algorithm.h.

## 6.161.2 Member Data Documentation

### 6.161.2.1 a

```
char Rcpp::algorithm::helpers::CTYPE_UNSIGNED_SHORT::a[11]
```

Definition at line 49 of file algorithm.h.

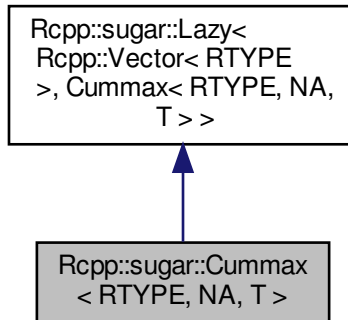
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

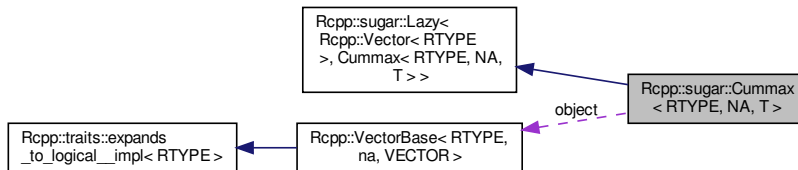
## 6.162 Rcpp::sugar::Cummax< RTYPE, NA, T > Class Template Reference

```
#include <cummax.h>
```

Inheritance diagram for Rcpp::sugar::Cummax< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Cummax< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::Vector< RTYPE >` `VECTOR`

### Public Member Functions

- `Cummax` (const `VEC_TYPE` &object\_)
- `VECTOR get` () const

## Private Attributes

- const [VEC\\_TYPE](#) & [object](#)

### 6.162.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::Cummax< RTYPE, NA, T >
```

Definition at line 29 of file cummax.h.

### 6.162.2 Member Typedef Documentation

#### 6.162.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Cummax< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file cummax.h.

#### 6.162.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE, NA, T> Rcpp::sugar::Cummax< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file cummax.h.

#### 6.162.2.3 VECTOR

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::Vector<RTYPE> Rcpp::sugar::Cummax< RTYPE, NA, T >::VECTOR
```

Definition at line 33 of file cummax.h.

### 6.162.3 Constructor & Destructor Documentation

### 6.162.3.1 Cummax()

```
template<int RTYPE, bool NA, typename T >  
Rcpp::sugar::Cummax< RTYPE, NA, T >::Cummax (   
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 35 of file cummax.h.

## 6.162.4 Member Function Documentation

### 6.162.4.1 get()

```
template<int RTYPE, bool NA, typename T >  
VECTOR Rcpp::sugar::Cummax< RTYPE, NA, T >::get ( ) const [inline]
```

Definition at line 37 of file cummax.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



## 6.162.5 Member Data Documentation

### 6.162.5.1 object

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::sugar::Cummax< RTYPE, NA, T >::object [private]
```

Definition at line 52 of file cummax.h.

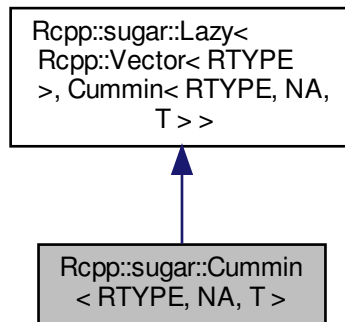
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cummax.h

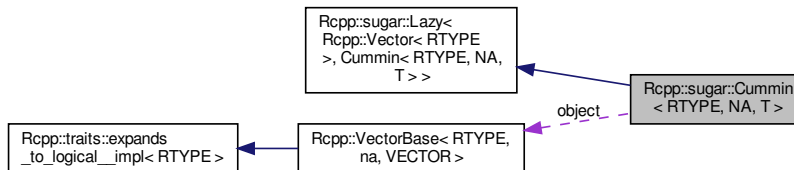
## 6.163 Rcpp::sugar::Cummin< RTYPE, NA, T > Class Template Reference

```
#include <cummin.h>
```

Inheritance diagram for Rcpp::sugar::Cummin< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Cummin< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::Vector< RTYPE >` `VECTOR`

### Public Member Functions

- `Cummin` (const `VEC_TYPE` &object\_)
- `VECTOR get` () const



## Private Attributes

- const [VEC\\_TYPE](#) & object

### 6.163.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::Cummin< RTYPE, NA, T >
```

Definition at line 29 of file cummin.h.

### 6.163.2 Member Typedef Documentation

#### 6.163.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Cummin< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file cummin.h.

#### 6.163.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE, NA, T> Rcpp::sugar::Cummin< RTYPE, NA, T >::VEC\_TYPE
```

Definition at line 31 of file cummin.h.

#### 6.163.2.3 VECTOR

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::Vector<RTYPE> Rcpp::sugar::Cummin< RTYPE, NA, T >::VECTOR
```

Definition at line 33 of file cummin.h.

### 6.163.3 Constructor & Destructor Documentation

### 6.163.3.1 Cummin()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Cummin< RTYPE, NA, T >::Cummin (
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 35 of file cummin.h.

## 6.163.4 Member Function Documentation

### 6.163.4.1 get()

```
template<int RTYPE, bool NA, typename T >
VECTOR Rcpp::sugar::Cummin< RTYPE, NA, T >::get ( ) const [inline]
```

Definition at line 37 of file cummin.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



## 6.163.5 Member Data Documentation

### 6.163.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Cummin< RTYPE, NA, T >::object [private]
```

Definition at line 52 of file cummin.h.

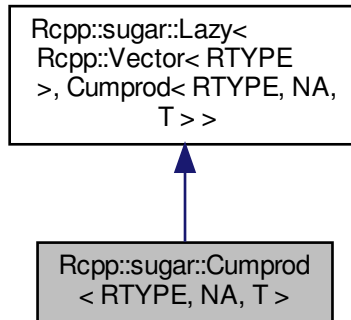
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/cummin.h`

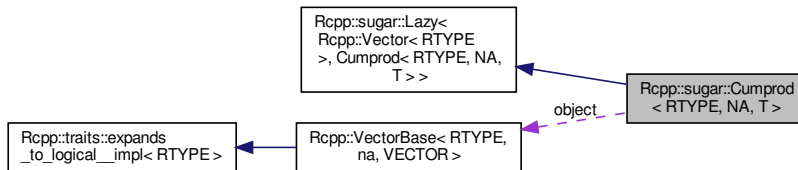
## 6.164 Rcpp::sugar::Cumprod< RTYPE, NA, T > Class Template Reference

```
#include <cumprod.h>
```

Inheritance diagram for Rcpp::sugar::Cumprod< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Cumprod< RTYPE, NA, T >:



### Public Types

- typedef [Rcpp::VectorBase< RTYPE, NA, T >](#) [VEC\\_TYPE](#)
- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) [STORAGE](#)
- typedef [Rcpp::Vector< RTYPE >](#) [VECTOR](#)

### Public Member Functions

- [Cumprod](#) (const [VEC\\_TYPE](#) &object\_)
- [VECTOR](#) [get](#) () const

## Private Attributes

- const [VEC\\_TYPE](#) & object

### 6.164.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::Cumprod< RTYPE, NA, T >
```

Definition at line 29 of file cumprod.h.

### 6.164.2 Member Typedef Documentation

#### 6.164.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Cumprod< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file cumprod.h.

#### 6.164.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE, NA, T> Rcpp::sugar::Cumprod< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file cumprod.h.

#### 6.164.2.3 VECTOR

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::Vector<RTYPE> Rcpp::sugar::Cumprod< RTYPE, NA, T >::VECTOR
```

Definition at line 33 of file cumprod.h.

### 6.164.3 Constructor & Destructor Documentation

### 6.164.3.1 Cumprod()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Cumprod< RTYPE, NA, T >::Cumprod (
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 35 of file cumprod.h.

## 6.164.4 Member Function Documentation

### 6.164.4.1 get()

```
template<int RTYPE, bool NA, typename T >
VECTOR Rcpp::sugar::Cumprod< RTYPE, NA, T >::get ( ) const [inline]
```

Definition at line 37 of file cumprod.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



## 6.164.5 Member Data Documentation

### 6.164.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Cumprod< RTYPE, NA, T >::object [private]
```

Definition at line 52 of file cumprod.h.

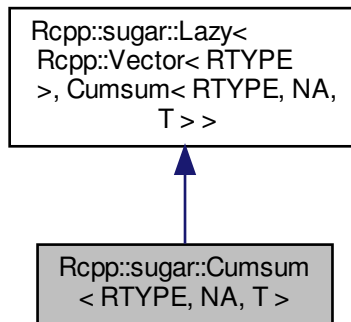
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/cumprod.h`

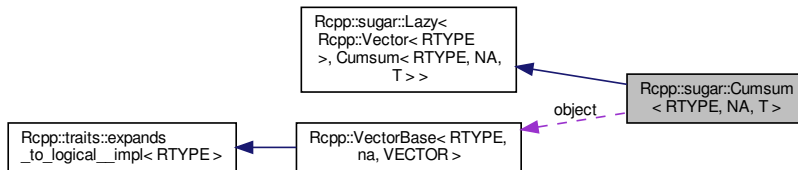
## 6.165 Rcpp::sugar::Cumsum< RTYPE, NA, T > Class Template Reference

```
#include <cumsum.h>
```

Inheritance diagram for Rcpp::sugar::Cumsum< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Cumsum< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::Vector< RTYPE >` `VECTOR`

### Public Member Functions

- `Cumsum` (const `VEC_TYPE` &object\_)
- `VECTOR get` () const

## Private Attributes

- const [VEC\\_TYPE](#) & object

### 6.165.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::Cumsum< RTYPE, NA, T >
```

Definition at line 29 of file cumsum.h.

### 6.165.2 Member Typedef Documentation

#### 6.165.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Cumsum< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file cumsum.h.

#### 6.165.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Cumsum< RTYPE, NA, T >::VEC\_TYPE
```

Definition at line 31 of file cumsum.h.

#### 6.165.2.3 VECTOR

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::Vector<RTYPE> Rcpp::sugar::Cumsum< RTYPE, NA, T >::VECTOR
```

Definition at line 33 of file cumsum.h.

### 6.165.3 Constructor & Destructor Documentation

### 6.165.3.1 Cumsum()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Cumsum< RTYPE, NA, T >::Cumsum (
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 35 of file cumsum.h.

## 6.165.4 Member Function Documentation

### 6.165.4.1 get()

```
template<int RTYPE, bool NA, typename T >
VECTOR Rcpp::sugar::Cumsum< RTYPE, NA, T >::get ( ) const [inline]
```

Definition at line 37 of file cumsum.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



## 6.165.5 Member Data Documentation

### 6.165.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Cumsum< RTYPE, NA, T >::object [private]
```

Definition at line 53 of file cumsum.h.

The documentation for this class was generated from the following file:

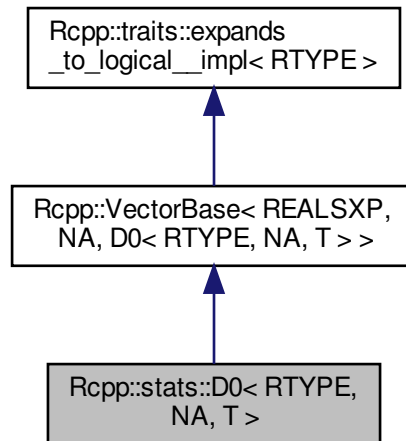
- `inst/include/Rcpp/sugar/functions/cumsum.h`



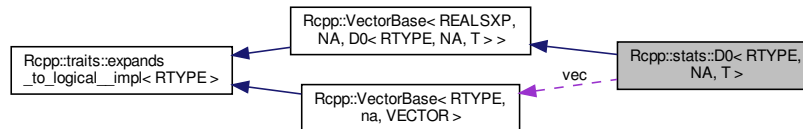
## 6.166 Rcpp::stats::D0< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::D0< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::D0< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `double(* FunPtr) (double, int)`

### Public Member Functions

- `D0 (FunPtr ptr_, const VEC_TYPE &vec_, bool log_)`
- `double operator[] (int i) const`
- `int size () const`

## Private Attributes

- [FunPtr ptr](#)
- const [VEC\\_TYPE](#) & [vec](#)
- int [log](#)

### 6.166.1 Detailed Description

```
template<int RTYPE, bool NA, typename T >
class Rcpp::stats::D0< RTYPE, NA, T >
```

Definition at line 33 of file dpq.h.

### 6.166.2 Member Typedef Documentation

#### 6.166.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(* Rcpp::stats::D0< RTYPE, NA, T >::FunPtr) (double, int)
```

Definition at line 36 of file dpq.h.

#### 6.166.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::D0< RTYPE, NA, T >::VEC\\_TYPE
```

Definition at line 35 of file dpq.h.

### 6.166.3 Constructor & Destructor Documentation

#### 6.166.3.1 D0()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::D0< RTYPE, NA, T >::D0 (
    FunPtr ptr\_,
    const VEC\_TYPE & vec\_,
    bool log\_ ) [inline]
```

Definition at line 38 of file dpq.h.

## 6.166.4 Member Function Documentation

### 6.166.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D0< RTYPE, NA, T >::operator[] (
    int i ) const [inline]
```

Definition at line 41 of file dpq.h.

References `Rcpp::stats::D0< RTYPE, NA, T >::log`, `Rcpp::stats::D0< RTYPE, NA, T >::ptr`, and `Rcpp::stats::D0< RTYPE, NA, T >::vec`.

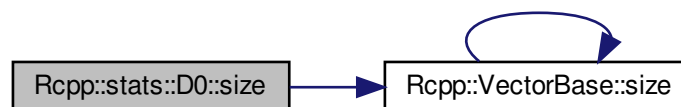
### 6.166.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::D0< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 45 of file dpq.h.

References `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`, and `Rcpp::stats::D0< RTYPE, NA, T >::vec`.

Here is the call graph for this function:



## 6.166.5 Member Data Documentation

### 6.166.5.1 log

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::D0< RTYPE, NA, T >::log [private]
```

Definition at line 50 of file dpq.h.

Referenced by Rcpp::stats::D0< RTYPE, NA, T >::operator[]().

### 6.166.5.2 ptr

```
template<int RTYPE, bool NA, typename T >  
FunPtr Rcpp::stats::D0< RTYPE, NA, T >::ptr [private]
```

Definition at line 48 of file dpq.h.

Referenced by Rcpp::stats::D0< RTYPE, NA, T >::operator[]().

### 6.166.5.3 vec

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::stats::D0< RTYPE, NA, T >::vec [private]
```

Definition at line 49 of file dpq.h.

Referenced by Rcpp::stats::D0< RTYPE, NA, T >::operator[](), and Rcpp::stats::D0< RTYPE, NA, T >::size().

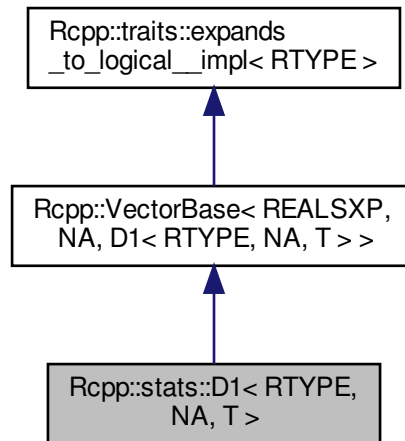
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/dpq/dpq.h](#)

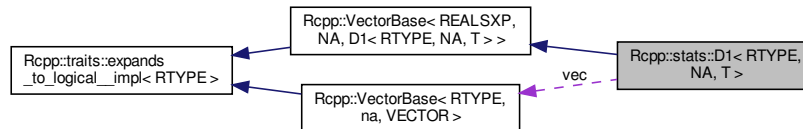
## 6.167 Rcpp::stats::D1< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::D1< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::D1< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `double(* FunPtr) (double, double, int)`

### Public Member Functions

- `D1 (FunPtr ptr_, const VEC_TYPE &vec_, double p0_, bool log_)`
- `double operator[] (int i) const`
- `int size () const`

## Private Attributes

- [FunPtr ptr](#)
- const [VEC\\_TYPE](#) & [vec](#)
- double [p0](#)
- int [log](#)

### 6.167.1 Detailed Description

```
template<int RTYPE, bool NA, typename T >  
class Rcpp::stats::D1< RTYPE, NA, T >
```

Definition at line 54 of file dpq.h.

### 6.167.2 Member Typedef Documentation

#### 6.167.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >  
typedef double(* Rcpp::stats::D1< RTYPE, NA, T >::FunPtr) (double, double, int)
```

Definition at line 57 of file dpq.h.

#### 6.167.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::D1< RTYPE, NA, T >::VEC\\_TYPE
```

Definition at line 56 of file dpq.h.

### 6.167.3 Constructor & Destructor Documentation

### 6.167.3.1 D1()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::D1< RTYPE, NA, T >::D1 (
    FunPtr ptr_,
    const VEC_TYPE & vec_,
    double p0_,
    bool log_ ) [inline]
```

Definition at line 59 of file dpq.h.

## 6.167.4 Member Function Documentation

### 6.167.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D1< RTYPE, NA, T >::operator[] (
    int i ) const [inline]
```

Definition at line 62 of file dpq.h.

References `Rcpp::stats::D1< RTYPE, NA, T >::log`, `Rcpp::stats::D1< RTYPE, NA, T >::p0`, `Rcpp::stats::D1< RTYPE, NA, T >::ptr`, and `Rcpp::stats::D1< RTYPE, NA, T >::vec`.

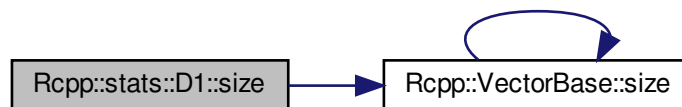
### 6.167.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::D1< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 66 of file dpq.h.

References `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`, and `Rcpp::stats::D1< RTYPE, NA, T >::vec`.

Here is the call graph for this function:



## 6.167.5 Member Data Documentation

### 6.167.5.1 log

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::D1< RTYPE, NA, T >::log [private]
```

Definition at line 72 of file dpq.h.

Referenced by Rcpp::stats::D1< RTYPE, NA, T >::operator[]().

### 6.167.5.2 p0

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::D1< RTYPE, NA, T >::p0 [private]
```

Definition at line 71 of file dpq.h.

Referenced by Rcpp::stats::D1< RTYPE, NA, T >::operator[]().

### 6.167.5.3 ptr

```
template<int RTYPE, bool NA, typename T >  
FunPtr Rcpp::stats::D1< RTYPE, NA, T >::ptr [private]
```

Definition at line 69 of file dpq.h.

Referenced by Rcpp::stats::D1< RTYPE, NA, T >::operator[]().

### 6.167.5.4 vec

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::stats::D1< RTYPE, NA, T >::vec [private]
```

Definition at line 70 of file dpq.h.

Referenced by Rcpp::stats::D1< RTYPE, NA, T >::operator[](), and Rcpp::stats::D1< RTYPE, NA, T >::size().

The documentation for this class was generated from the following file:

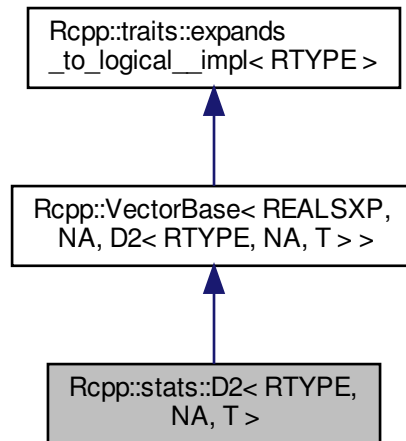
- inst/include/Rcpp/stats/dpq/dpq.h



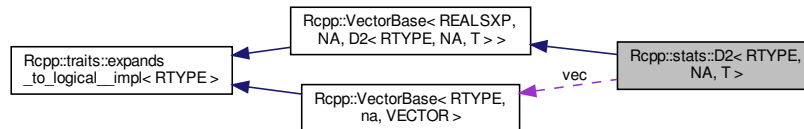
## 6.168 Rcpp::stats::D2< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::D2< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::D2< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `double(* FunPtr) (double, double, double, int)`

### Public Member Functions

- `D2 (FunPtr ptr_, const VEC_TYPE &vec_, double p0_, double p1_, bool log_)`
- `double operator[] (int i) const`
- `int size () const`

## Private Attributes

- [FunPtr ptr](#)
- const [VEC\\_TYPE](#) & [vec](#)
- double [p0](#)
- double [p1](#)
- int [log](#)

### 6.168.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::stats::D2< RTYPE, NA, T >
```

Definition at line 76 of file dpq.h.

### 6.168.2 Member Typedef Documentation

#### 6.168.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >  
typedef double(* Rcpp::stats::D2< RTYPE, NA, T >::FunPtr) (double, double, double, int)
```

Definition at line 79 of file dpq.h.

#### 6.168.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::D2< RTYPE, NA, T >::VEC\_TYPE
```

Definition at line 78 of file dpq.h.

### 6.168.3 Constructor & Destructor Documentation

### 6.168.3.1 D2()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::D2< RTYPE, NA, T >::D2 (
    FunPtr ptr_,
    const VEC_TYPE & vec_,
    double p0_,
    double p1_,
    bool log_ ) [inline]
```

Definition at line 81 of file dpq.h.

## 6.168.4 Member Function Documentation

### 6.168.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D2< RTYPE, NA, T >::operator[] (
    int i ) const [inline]
```

Definition at line 84 of file dpq.h.

References `Rcpp::stats::D2< RTYPE, NA, T >::log`, `Rcpp::stats::D2< RTYPE, NA, T >::p0`, `Rcpp::stats::D2< RTYPE, NA, T >::p1`, `Rcpp::stats::D2< RTYPE, NA, T >::ptr`, and `Rcpp::stats::D2< RTYPE, NA, T >::vec`.

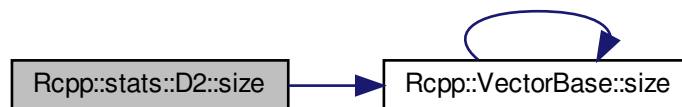
### 6.168.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::D2< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 88 of file dpq.h.

References `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`, and `Rcpp::stats::D2< RTYPE, NA, T >::vec`.

Here is the call graph for this function:



## 6.168.5 Member Data Documentation

### 6.168.5.1 log

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::D2< RTYPE, NA, T >::log [private]
```

Definition at line 94 of file dpq.h.

Referenced by Rcpp::stats::D2< RTYPE, NA, T >::operator[]().

### 6.168.5.2 p0

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::D2< RTYPE, NA, T >::p0 [private]
```

Definition at line 93 of file dpq.h.

Referenced by Rcpp::stats::D2< RTYPE, NA, T >::operator[]().

### 6.168.5.3 p1

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::D2< RTYPE, NA, T >::p1 [private]
```

Definition at line 93 of file dpq.h.

Referenced by Rcpp::stats::D2< RTYPE, NA, T >::operator[]().

### 6.168.5.4 ptr

```
template<int RTYPE, bool NA, typename T >  
FunPtr Rcpp::stats::D2< RTYPE, NA, T >::ptr [private]
```

Definition at line 91 of file dpq.h.

Referenced by Rcpp::stats::D2< RTYPE, NA, T >::operator[]().

## 6.168.5.5 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::D2< RTYPE, NA, T >::vec [private]
```

Definition at line 92 of file dpq.h.

Referenced by Rcpp::stats::D2< RTYPE, NA, T >::operator[](), and Rcpp::stats::D2< RTYPE, NA, T >::size().

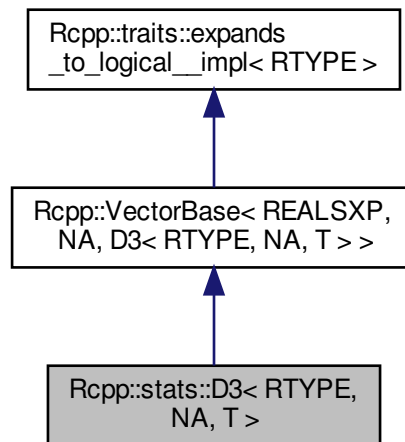
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

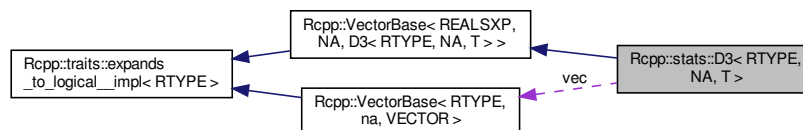
## 6.169 Rcpp::stats::D3&lt; RTYPE, NA, T &gt; Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::D3< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::D3< RTYPE, NA, T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, [NA](#), T > [VEC\\_TYPE](#)
- typedef double(\* [FunPtr](#)) (double, double, double, double, int)

## Public Member Functions

- [D3](#) ([FunPtr](#) ptr\_, const [VEC\\_TYPE](#) &vec\_, double p0\_, double p1\_, double p2\_, bool log\_)
- double [operator\[\]](#) (int i) const
- int [size](#) () const

## Private Attributes

- [FunPtr](#) ptr
- const [VEC\\_TYPE](#) & [vec](#)
- double [p0](#)
- double [p1](#)
- double [p2](#)
- int [log](#)

### 6.169.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>
class Rcpp::stats::D3< RTYPE, NA, T >
```

Definition at line 98 of file dpq.h.

### 6.169.2 Member Typedef Documentation

#### 6.169.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >
typedef double(* Rcpp::stats::D3< RTYPE, NA, T >::FunPtr) (double, double, double, double, int)
```

Definition at line 101 of file dpq.h.

### 6.169.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::D3< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 100 of file dpq.h.

## 6.169.3 Constructor & Destructor Documentation

### 6.169.3.1 D3()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::D3< RTYPE, NA, T >::D3 (
    FunPtr ptr_,
    const VEC_TYPE & vec_,
    double p0_,
    double p1_,
    double p2_,
    bool log_ ) [inline]
```

Definition at line 103 of file dpq.h.

## 6.169.4 Member Function Documentation

### 6.169.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D3< RTYPE, NA, T >::operator[] (
    int i ) const [inline]
```

Definition at line 106 of file dpq.h.

References [Rcpp::stats::D3< RTYPE, NA, T >::log](#), [Rcpp::stats::D3< RTYPE, NA, T >::p0](#), [Rcpp::stats::D3< RTYPE, NA, T >::p1](#), [Rcpp::stats::D3< RTYPE, NA, T >::p2](#), [Rcpp::stats::D3< RTYPE, NA, T >::ptr](#), and [Rcpp::stats::D3< RTYPE, NA, T >::vec](#).

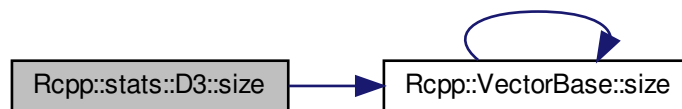
### 6.169.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::D3< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 110 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::D3< RTYPE, NA, T >::vec.

Here is the call graph for this function:



## 6.169.5 Member Data Documentation

### 6.169.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::D3< RTYPE, NA, T >::log [private]
```

Definition at line 116 of file dpq.h.

Referenced by Rcpp::stats::D3< RTYPE, NA, T >::operator[]().

### 6.169.5.2 p0

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::D3< RTYPE, NA, T >::p0 [private]
```

Definition at line 115 of file dpq.h.

Referenced by Rcpp::stats::D3< RTYPE, NA, T >::operator[]().



### 6.169.5.3 p1

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::D3< RTYPE, NA, T >::p1 [private]
```

Definition at line 115 of file dpq.h.

Referenced by Rcpp::stats::D3< RTYPE, NA, T >::operator[]().

### 6.169.5.4 p2

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::D3< RTYPE, NA, T >::p2 [private]
```

Definition at line 115 of file dpq.h.

Referenced by Rcpp::stats::D3< RTYPE, NA, T >::operator[]().

### 6.169.5.5 ptr

```
template<int RTYPE, bool NA, typename T >  
FunPtr Rcpp::stats::D3< RTYPE, NA, T >::ptr [private]
```

Definition at line 113 of file dpq.h.

Referenced by Rcpp::stats::D3< RTYPE, NA, T >::operator[]().

### 6.169.5.6 vec

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::stats::D3< RTYPE, NA, T >::vec [private]
```

Definition at line 114 of file dpq.h.

Referenced by Rcpp::stats::D3< RTYPE, NA, T >::operator[](), and Rcpp::stats::D3< RTYPE, NA, T >::size().

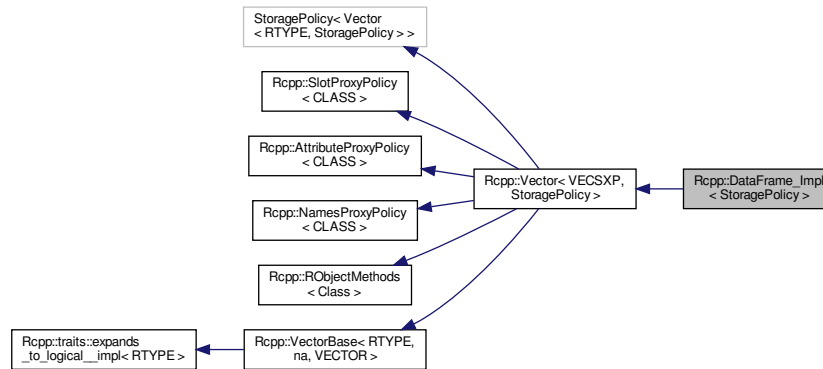
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/dpq/dpq.h](#)

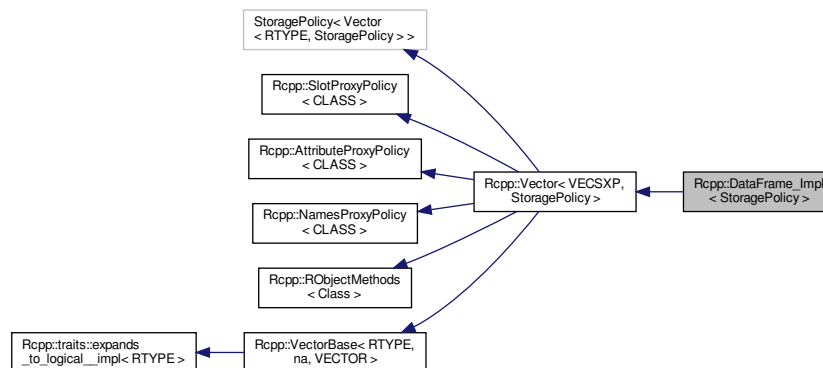
## 6.170 Rcpp::DataFrame\_Impl< StoragePolicy > Class Template Reference

```
#include <DataFrame.h>
```

Inheritance diagram for Rcpp::DataFrame\_Impl< StoragePolicy >:



Collaboration diagram for Rcpp::DataFrame\_Impl< StoragePolicy >:



### Public Types

- typedef [Vector](#)< VECSXP, StoragePolicy > [Parent](#)

### Public Member Functions

- [DataFrame\\_Impl](#) ()
- [DataFrame\\_Impl](#) (SEXP x)

- [DataFrame\\_Impl](#) (const [DataFrame\\_Impl](#) &other)
- template<typename T >  
[DataFrame\\_Impl](#) (const T &obj)
- [DataFrame\\_Impl](#) & operator= ([DataFrame\\_Impl](#) &other)
- [DataFrame\\_Impl](#) & operator= (SEXP x)
- int [nrow](#) () const
- template<typename T >  
void [push\\_back](#) (const T &object)
- template<typename T >  
void [push\\_back](#) (const T &object, const std::string &name)
- template<typename T >  
void [push\\_front](#) (const T &object)
- template<typename T >  
void [push\\_front](#) (const T &object, const std::string &name)
- int [nrows](#) () const
- int [rows](#) () const
- R\_xlen\_t [ncol](#) () const
- R\_xlen\_t [cols](#) () const

### Static Public Member Functions

- static [DataFrame\\_Impl](#) [create](#) ()

### Private Member Functions

- void [set\\_\\_](#) (SEXP x)
- void [set\\_type\\_after\\_push](#) ()

### Static Private Member Functions

- static [DataFrame\\_Impl](#) [from\\_list](#) ([Parent](#) obj)

### Additional Inherited Members

#### 6.170.1 Detailed Description

```
template<template< class > class StoragePolicy>
class Rcpp::DataFrame_Impl< StoragePolicy >
```

Definition at line 38 of file [DataFrame.h](#).

#### 6.170.2 Member Typedef Documentation

6.170.2.1 Parent

```
template<template< class > class StoragePolicy>
typedef Vector<VECSXP, StoragePolicy> Rcpp::DataFrame_Impl< StoragePolicy >::Parent
```

Definition at line 40 of file DataFrame.h.

6.170.3 Constructor & Destructor Documentation

6.170.3.1 DataFrame\_Impl() [1/4]

```
template<template< class > class StoragePolicy>
Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl ( ) [inline]
```

Definition at line 42 of file DataFrame.h.

Referenced by Rcpp::DataFrame\_Impl< StoragePolicy >::create(), and Rcpp::DataFrame\_Impl< StoragePolicy >::from\_list().

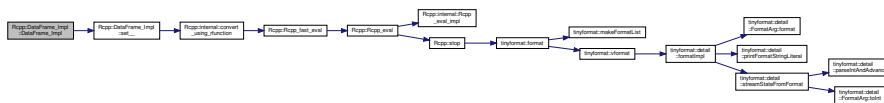
6.170.3.2 DataFrame\_Impl() [2/4]

```
template<template< class > class StoragePolicy>
Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl (
    SEXP x ) [inline]
```

Definition at line 43 of file DataFrame.h.

References Rcpp::DataFrame\_Impl< StoragePolicy >::set\_\_().

Here is the call graph for this function:



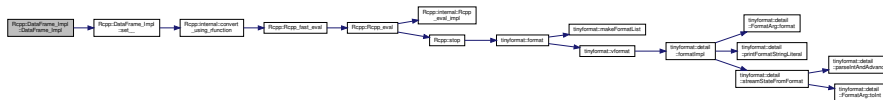
**6.170.3.3 DataFrame\_Impl()** [3/4]

```
template<template< class > class StoragePolicy>
Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl (
    const DataFrame_Impl< StoragePolicy > & other ) [inline]
```

Definition at line 46 of file DataFrame.h.

References Rcpp::DataFrame\_Impl< StoragePolicy >::set\_\_().

Here is the call graph for this function:

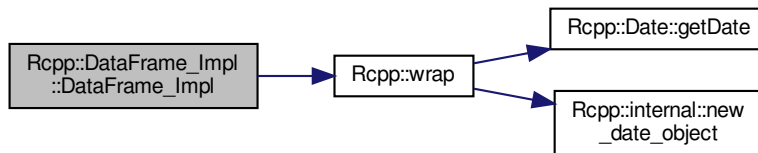
**6.170.3.4 DataFrame\_Impl()** [4/4]

```
template<template< class > class StoragePolicy>
template<class T >
Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl (
    const T & obj )
```

Definition at line 25 of file DataFrame.h.

References Rcpp::wrap().

Here is the call graph for this function:

**6.170.4 Member Function Documentation**

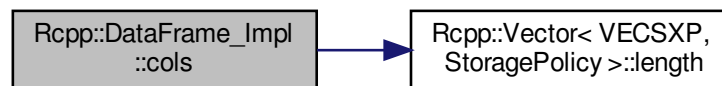
### 6.170.4.1 cols()

```
template<template< class > class StoragePolicy>  
R_xlen_t Rcpp::DataFrame_Impl< StoragePolicy >::cols ( ) const [inline]
```

Definition at line 115 of file DataFrame.h.

References Rcpp::Vector< VECSXP, StoragePolicy >::length().

Here is the call graph for this function:



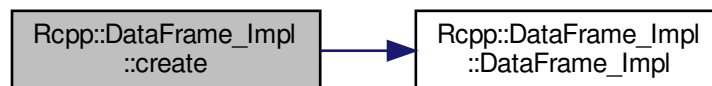
### 6.170.4.2 create()

```
template<template< class > class StoragePolicy>  
static DataFrame_Impl Rcpp::DataFrame_Impl< StoragePolicy >::create ( ) [inline], [static]
```

Definition at line 117 of file DataFrame.h.

References Rcpp::DataFrame\_Impl< StoragePolicy >::DataFrame\_Impl().

Here is the call graph for this function:





### 6.170.4.5 nrow()

```
template<template< class > class StoragePolicy>
int Rcpp::DataFrame_Impl< StoragePolicy >::nrow ( ) const [inline]
```

Definition at line 69 of file DataFrame.h.

Referenced by Rcpp::DataFrame\_Impl< StoragePolicy >::nrows(), and Rcpp::DataFrame\_Impl< StoragePolicy >←::rows().

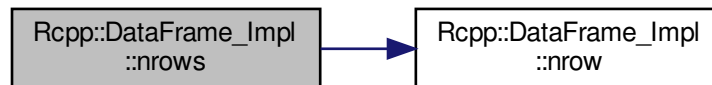
### 6.170.4.6 nrows()

```
template<template< class > class StoragePolicy>
int Rcpp::DataFrame_Impl< StoragePolicy >::nrows ( ) const [inline]
```

Definition at line 111 of file DataFrame.h.

References Rcpp::DataFrame\_Impl< StoragePolicy >::nrow().

Here is the call graph for this function:



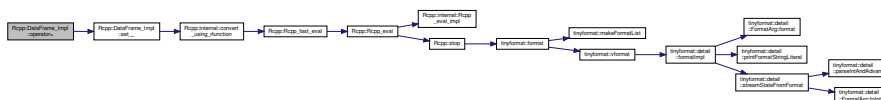
### 6.170.4.7 operator=() [1/2]

```
template<template< class > class StoragePolicy>
DataFrame_Impl& Rcpp::DataFrame_Impl< StoragePolicy >::operator= (
    DataFrame_Impl< StoragePolicy > & other ) [inline]
```

Definition at line 53 of file DataFrame.h.

References Rcpp::DataFrame\_Impl< StoragePolicy >::set\_\_().

Here is the call graph for this function:





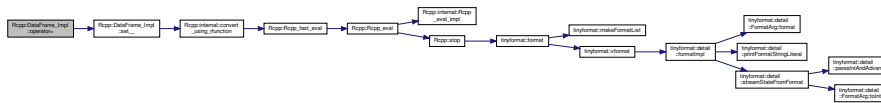
## 6.170.4.8 operator=() [2/2]

```
template<template< class > class StoragePolicy>
DataFrame_Impl& Rcpp::DataFrame_Impl< StoragePolicy >::operator= (
    SEXP x ) [inline]
```

Definition at line 58 of file DataFrame.h.

References Rcpp::DataFrame\_Impl< StoragePolicy >::set\_\_().

Here is the call graph for this function:



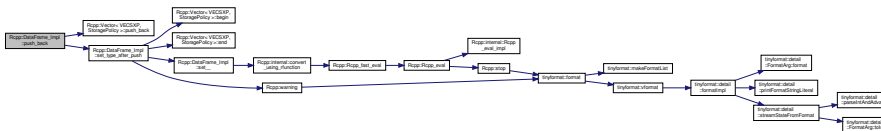
## 6.170.4.9 push\_back() [1/2]

```
template<template< class > class StoragePolicy>
template<typename T >
void Rcpp::DataFrame_Impl< StoragePolicy >::push_back (
    const T & object ) [inline]
```

Definition at line 87 of file DataFrame.h.

References Rcpp::Vector< VECSXP, StoragePolicy >::push\_back(), and Rcpp::DataFrame\_Impl< StoragePolicy >::set\_type\_after\_push().

Here is the call graph for this function:



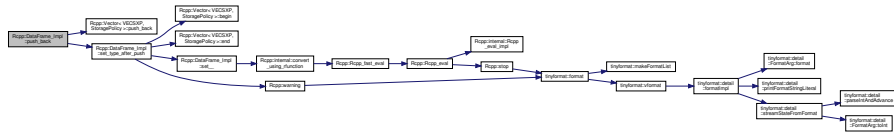
### 6.170.4.10 push\_back() [2/2]

```
template<template< class > class StoragePolicy>  
template<typename T >  
void Rcpp::DataFrame_Impl< StoragePolicy >::push_back (  
    const T & object,  
    const std::string & name ) [inline]
```

Definition at line 93 of file DataFrame.h.

References Rcpp::Vector< VECSXP, StoragePolicy >::push\_back(), and Rcpp::DataFrame\_Impl< StoragePolicy >::set\_type\_after\_push().

Here is the call graph for this function:



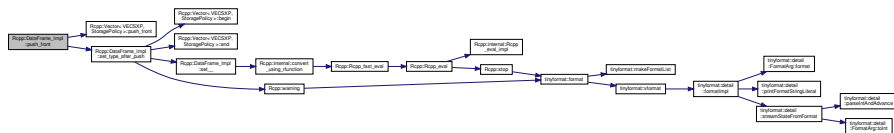
### 6.170.4.11 push\_front() [1/2]

```
template<template< class > class StoragePolicy>  
template<typename T >  
void Rcpp::DataFrame_Impl< StoragePolicy >::push_front (  
    const T & object ) [inline]
```

Definition at line 99 of file DataFrame.h.

References Rcpp::Vector< VECSXP, StoragePolicy >::push\_front(), and Rcpp::DataFrame\_Impl< StoragePolicy >::set\_type\_after\_push().

Here is the call graph for this function:



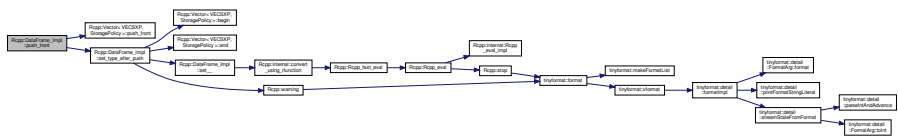
**6.170.4.12 push\_front() [2/2]**

```
template<template< class > class StoragePolicy>
template<typename T >
void Rcpp::DataFrame_Impl< StoragePolicy >::push_front (
    const T & object,
    const std::string & name ) [inline]
```

Definition at line 105 of file DataFrame.h.

References Rcpp::Vector< VECSXP, StoragePolicy >::push\_front(), and Rcpp::DataFrame\_Impl< StoragePolicy >↔::set\_type\_after\_push().

Here is the call graph for this function:

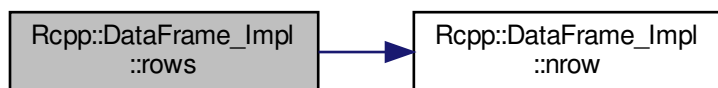
**6.170.4.13 rows()**

```
template<template< class > class StoragePolicy>
int Rcpp::DataFrame_Impl< StoragePolicy >::rows ( ) const [inline]
```

Definition at line 112 of file DataFrame.h.

References Rcpp::DataFrame\_Impl< StoragePolicy >::nrow().

Here is the call graph for this function:



### 6.170.4.14 `set__()`

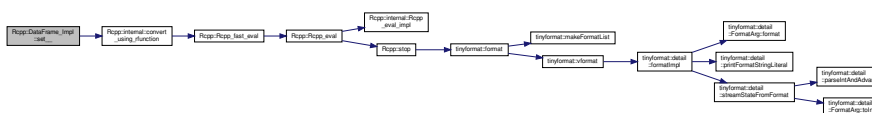
```
template<template< class > class StoragePolicy>
void Rcpp::DataFrame_Impl< StoragePolicy >::set__ (
    SEXP x ) [inline], [private]
```

Definition at line 124 of file `DataFrame.h`.

References `Rcpp::internal::convert_using_rfunction()`.

Referenced by `Rcpp::DataFrame_Impl< StoragePolicy >::DataFrame_Impl()`, `Rcpp::DataFrame_Impl< StoragePolicy >::operator=()`, and `Rcpp::DataFrame_Impl< StoragePolicy >::set_type_after_push()`.

Here is the call graph for this function:



### 6.170.4.15 `set_type_after_push()`

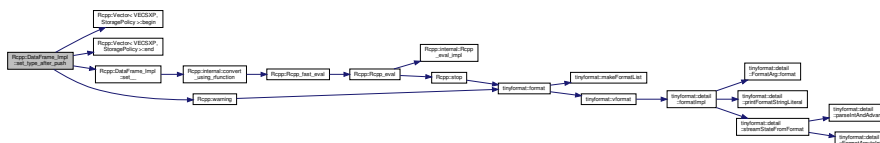
```
template<template< class > class StoragePolicy>
void Rcpp::DataFrame_Impl< StoragePolicy >::set_type_after_push ( ) [inline], [private]
```

Definition at line 133 of file `DataFrame.h`.

References `Rcpp::Vector< VECSXP, StoragePolicy >::begin()`, `Rcpp::Vector< VECSXP, StoragePolicy >::end()`, `Rcpp::DataFrame_Impl< StoragePolicy >::set__()`, and `Rcpp::warning()`.

Referenced by `Rcpp::DataFrame_Impl< StoragePolicy >::push_back()`, and `Rcpp::DataFrame_Impl< StoragePolicy >::push_front()`.

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- `inst/include/Rcpp/`[DataFrame.h](#)

## 6.171 Rcpp::Date Class Reference

```
#include <Date.h>
```

### Public Member Functions

- [Date](#) ()
- [Date](#) (SEXP s)
- [Date](#) (const int &dt)
- [Date](#) (const double &dt)
- [Date](#) (const std::string &s, const std::string &fmt="%Y-%m-%d")
- [Date](#) (const unsigned int &mon, const unsigned int &day, const unsigned int &year)
- double [getDate](#) (void) const
- int [getDay](#) () const
- int [getMonth](#) () const
- int [getYear](#) () const
- int [getWeekday](#) () const
- int [getYearday](#) () const
- int [is\\_na](#) () const
- [operator double](#) () const
- std::string [format](#) (const char \*fmt="%Y-%m-%d") const

### Static Public Member Functions

- static unsigned int [baseYear](#) ()

### Private Member Functions

- void [update\\_tm](#) ()

### Private Attributes

- double [m\\_d](#)
- struct [tm m\\_tm](#)

### Friends

- [Date operator+](#) (const [Date](#) &date, int offset)
- double [operator-](#) (const [Date](#) &date1, const [Date](#) &date2)
- bool [operator<](#) (const [Date](#) &date1, const [Date](#) &date2)
- bool [operator>](#) (const [Date](#) &date1, const [Date](#) &date2)
- bool [operator==](#) (const [Date](#) &date1, const [Date](#) &date2)
- bool [operator>=](#) (const [Date](#) &date1, const [Date](#) &date2)
- bool [operator<=](#) (const [Date](#) &date1, const [Date](#) &date2)
- bool [operator!=](#) (const [Date](#) &date1, const [Date](#) &date2)
- std::ostream & [operator<<](#) (std::ostream &os, const [Date](#) d)

### 6.171.1 Detailed Description

Definition at line 31 of file Date.h.

### 6.171.2 Constructor & Destructor Documentation

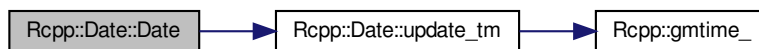
#### 6.171.2.1 Date() [1/6]

```
Rcpp::Date::Date ( ) [inline]
```

Definition at line 33 of file Date.h.

References `m_d`, and `update_tm()`.

Here is the call graph for this function:



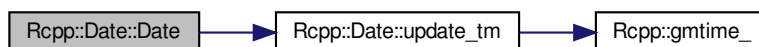
#### 6.171.2.2 Date() [2/6]

```
Rcpp::Date::Date (
    SEXP s ) [inline]
```

Definition at line 25 of file Date.h.

References `m_d`, and `update_tm()`.

Here is the call graph for this function:



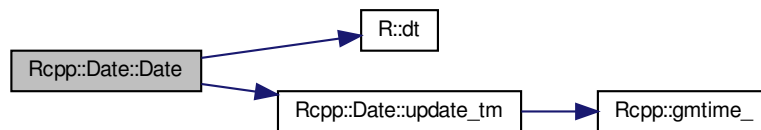
### 6.171.2.3 Date() [3/6]

```
Rcpp::Date::Date (  
    const int & dt ) [inline]
```

Definition at line 40 of file Date.h.

References R::dt(), m\_d, and update\_tm().

Here is the call graph for this function:



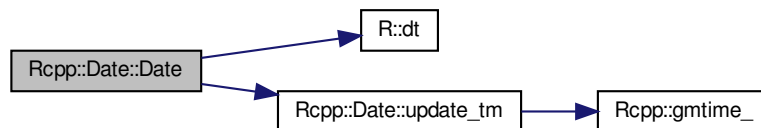
### 6.171.2.4 Date() [4/6]

```
Rcpp::Date::Date (  
    const double & dt ) [inline]
```

Definition at line 46 of file Date.h.

References R::dt(), m\_d, and update\_tm().

Here is the call graph for this function:



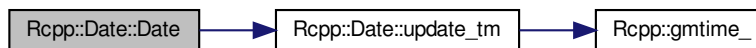
**6.171.2.5 Date()** [5/6]

```
Rcpp::Date::Date (
    const std::string & s,
    const std::string & fmt = "%Y-%m-%d" ) [inline]
```

Definition at line 30 of file Date.h.

References `m_d`, and `update_tm()`.

Here is the call graph for this function:

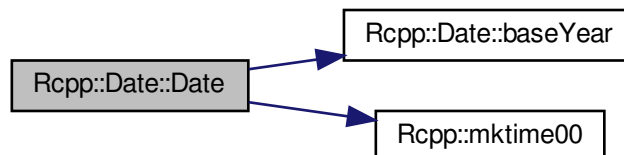
**6.171.2.6 Date()** [6/6]

```
Rcpp::Date::Date (
    const unsigned int & mon,
    const unsigned int & day,
    const unsigned int & year ) [inline]
```

Definition at line 52 of file Date.h.

References `baseYear()`, `m_d`, `m_tm`, and `Rcpp::mktime00()`.

Here is the call graph for this function:





## 6.171.3 Member Function Documentation

### 6.171.3.1 baseYear()

```
static unsigned int Rcpp::Date::baseYear ( ) [inline], [static]
```

Definition at line 85 of file Date.h.

Referenced by Date(), and format().

### 6.171.3.2 format()

```
std::string Rcpp::Date::format (
    const char * fmt = "%Y-%m-%d" ) const [inline]
```

Definition at line 107 of file Date.h.

References baseYear(), m\_tm, and Rcpp::tm.

Here is the call graph for this function:



### 6.171.3.3 getDate()

```
double Rcpp::Date::getDate (
    void ) const [inline]
```

Definition at line 70 of file Date.h.

References m\_d.

Referenced by Rcpp::internal::caster< Rcpp::Date, double >(), and Rcpp::wrap().

#### 6.171.3.4 `getDay()`

```
int Rcpp::Date::getDay ( ) const [inline]
```

Definition at line 78 of file Date.h.

References `m_tm`.

#### 6.171.3.5 `getMonth()`

```
int Rcpp::Date::getMonth ( ) const [inline]
```

Definition at line 79 of file Date.h.

References `m_tm`.

#### 6.171.3.6 `getWeekday()`

```
int Rcpp::Date::getWeekday ( ) const [inline]
```

Definition at line 81 of file Date.h.

References `m_tm`.

#### 6.171.3.7 `getYear()`

```
int Rcpp::Date::getYear ( ) const [inline]
```

Definition at line 80 of file Date.h.

References `m_tm`.

#### 6.171.3.8 `getYearday()`

```
int Rcpp::Date::getYearday ( ) const [inline]
```

Definition at line 82 of file Date.h.

References `m_tm`.

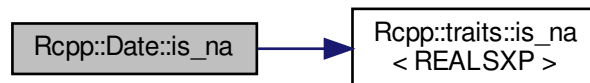
### 6.171.3.9 is\_na()

```
int Rcpp::Date::is_na ( ) const [inline]
```

Definition at line 99 of file Date.h.

References Rcpp::traits::is\_na< REALSXP >(), and m\_d.

Here is the call graph for this function:



### 6.171.3.10 operator double()

```
Rcpp::Date::operator double ( ) const [inline]
```

Definition at line 103 of file Date.h.

References m\_d.

### 6.171.3.11 update\_tm()

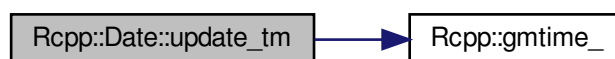
```
void Rcpp::Date::update_tm ( ) [inline], [private]
```

Definition at line 126 of file Date.h.

References Rcpp::gmtime\_(), m\_d, and m\_tm.

Referenced by Date().

Here is the call graph for this function:



## 6.171.4 Friends And Related Function Documentation

### 6.171.4.1 operator"!="

```
bool operator!= (
    const Date & date1,
    const Date & date2 ) [friend]
```

Definition at line 170 of file Date.h.

### 6.171.4.2 operator+

```
Date operator+ (
    const Date & date,
    int offset ) [friend]
```

Definition at line 156 of file Date.h.

### 6.171.4.3 operator-

```
double operator- (
    const Date & date1,
    const Date & date2 ) [friend]
```

Definition at line 164 of file Date.h.

### 6.171.4.4 operator<

```
bool operator< (
    const Date & date1,
    const Date & date2 ) [friend]
```

Definition at line 165 of file Date.h.

#### 6.171.4.5 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const Date d ) [friend]
```

Definition at line 172 of file Date.h.

#### 6.171.4.6 operator<=

```
bool operator<= (
    const Date & date1,
    const Date & date2 ) [friend]
```

Definition at line 169 of file Date.h.

#### 6.171.4.7 operator==

```
bool operator== (
    const Date & date1,
    const Date & date2 ) [friend]
```

Definition at line 167 of file Date.h.

#### 6.171.4.8 operator>

```
bool operator> (
    const Date & date1,
    const Date & date2 ) [friend]
```

Definition at line 166 of file Date.h.

#### 6.171.4.9 operator>=

```
bool operator>= (
    const Date & date1,
    const Date & date2 ) [friend]
```

Definition at line 168 of file Date.h.

## 6.171.5 Member Data Documentation

### 6.171.5.1 m\_d

```
double Rcpp::Date::m_d [private]
```

Definition at line 122 of file Date.h.

Referenced by Date(), getDate(), is\_na(), operator double(), and update\_tm().

### 6.171.5.2 m\_tm

```
struct tm Rcpp::Date::m_tm [private]
```

Definition at line 122 of file Date.h.

Referenced by Date(), format(), getDay(), getMonth(), getWeekday(), getYear(), getYearday(), and update\_tm().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/date\_datetime/Date.h

## 6.172 Rcpp::Datetime Class Reference

```
#include <Datetime.h>
```

### Public Member Functions

- [Datetime](#) ()
- [Datetime](#) (SEXP s)
- [Datetime](#) (const double &dt)
- [Datetime](#) (const std::string &s, const std::string &fmt="%Y-%m-%d %H:%M:%OS")
- double [getFractionalTimestamp](#) (void) const
- int [getMicroSeconds](#) () const
- int [getSeconds](#) () const
- int [getMinutes](#) () const
- int [getHours](#) () const
- int [getDay](#) () const
- int [getMonth](#) () const
- int [getYear](#) () const
- int [getWeekday](#) () const
- int [getYearday](#) () const
- int [is\\_na](#) () const
- [operator double](#) () const
- std::string [format](#) (const char \*fmt="%Y-%m-%d %H:%M:%S") const

## Private Member Functions

- void [update\\_tm](#) ()

## Static Private Member Functions

- static unsigned int [baseYear](#) ()

## Private Attributes

- double [m\\_dt](#)
- struct [tm](#) [m\\_tm](#)
- unsigned int [m\\_us](#)

## Friends

- [Datetime operator+](#) (const [Datetime](#) &dt, double offset)
- [Datetime operator+](#) (const [Datetime](#) &dt, int offset)
- double [operator-](#) (const [Datetime](#) &dt1, const [Datetime](#) &dt2)
- bool [operator<](#) (const [Datetime](#) &dt1, const [Datetime](#) &dt2)
- bool [operator>](#) (const [Datetime](#) &dt1, const [Datetime](#) &dt2)
- bool [operator==](#) (const [Datetime](#) &dt1, const [Datetime](#) &dt2)
- bool [operator>=](#) (const [Datetime](#) &dt1, const [Datetime](#) &dt2)
- bool [operator<=](#) (const [Datetime](#) &dt1, const [Datetime](#) &dt2)
- bool [operator!=](#) (const [Datetime](#) &dt1, const [Datetime](#) &dt2)
- std::ostream & [operator<<](#) (std::ostream &s, const [Datetime](#) d)

### 6.172.1 Detailed Description

Definition at line 33 of file [Datetime.h](#).

### 6.172.2 Constructor & Destructor Documentation

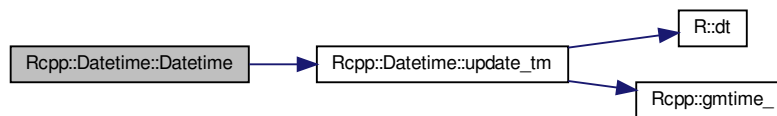
### 6.172.2.1 Datetime() [1/4]

```
Rcpp::Datetime::Datetime ( ) [inline]
```

Definition at line 35 of file Datetime.h.

References `m_dt`, and `update_tm()`.

Here is the call graph for this function:



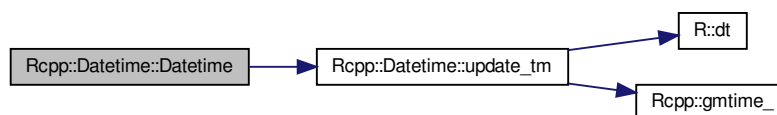
### 6.172.2.2 Datetime() [2/4]

```
Rcpp::Datetime::Datetime (
    SEXP s ) [inline]
```

Definition at line 25 of file Datetime.h.

References `m_dt`, and `update_tm()`.

Here is the call graph for this function:





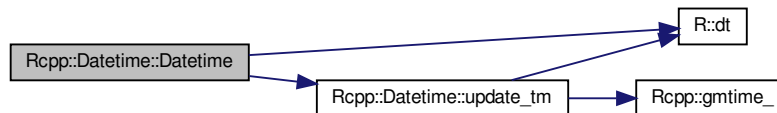
### 6.172.2.3 Datetime() [3/4]

```
Rcpp::Datetime::Datetime (
    const double & dt ) [inline]
```

Definition at line 42 of file Datetime.h.

References R::dt(), m\_dt, and update\_tm().

Here is the call graph for this function:



### 6.172.2.4 Datetime() [4/4]

```
Rcpp::Datetime::Datetime (
    const std::string & s,
    const std::string & fmt = "%Y-%m-%d %H:%M:%OS" ) [inline]
```

Definition at line 30 of file Datetime.h.

References m\_dt, and update\_tm().

Here is the call graph for this function:



## 6.172.3 Member Function Documentation

### 6.172.3.1 baseYear()

```
static unsigned int Rcpp::Datetime::baseYear ( ) [inline], [static], [private]
```

Definition at line 116 of file Datetime.h.

### 6.172.3.2 format()

```
std::string Rcpp::Datetime::format (
    const char * fmt = "%Y-%m-%d %H:%M:%S" ) const [inline]
```

Definition at line 75 of file Datetime.h.

References `m_dt`, `m_us`, and `Rcpp::tm`.

### 6.172.3.3 getDay()

```
int Rcpp::Datetime::getDay ( ) const [inline]
```

Definition at line 54 of file Datetime.h.

References `m_tm`.

### 6.172.3.4 getFractionalTimestamp()

```
double Rcpp::Datetime::getFractionalTimestamp (
    void ) const [inline]
```

Definition at line 48 of file Datetime.h.

References `m_dt`.

Referenced by `Rcpp::internal::caster< Rcpp::Datetime, double >()`, and `Rcpp::wrap< Datetime >()`.

### 6.172.3.5 getHours()

```
int Rcpp::Datetime::getHours ( ) const [inline]
```

Definition at line 53 of file Datetime.h.

References `m_tm`.

### 6.172.3.6 getMicroSeconds()

```
int Rcpp::Datetime::getMicroSeconds ( ) const [inline]
```

Definition at line 50 of file Datetime.h.

References `m_us`.

### 6.172.3.7 getMinutes()

```
int Rcpp::Datetime::getMinutes ( ) const [inline]
```

Definition at line 52 of file Datetime.h.

References `m_tm`.

### 6.172.3.8 getMonth()

```
int Rcpp::Datetime::getMonth ( ) const [inline]
```

Definition at line 55 of file Datetime.h.

References `m_tm`.

### 6.172.3.9 getSeconds()

```
int Rcpp::Datetime::getSeconds ( ) const [inline]
```

Definition at line 51 of file Datetime.h.

References `m_tm`.

### 6.172.3.10 getWeekday()

```
int Rcpp::Datetime::getWeekday ( ) const [inline]
```

Definition at line 57 of file Datetime.h.

References `m_tm`.

### 6.172.3.11 getYear()

```
int Rcpp::Datetime::getYear ( ) const [inline]
```

Definition at line 56 of file Datetime.h.

References `m_tm`.

### 6.172.3.12 getYearday()

```
int Rcpp::Datetime::getYearday ( ) const [inline]
```

Definition at line 58 of file Datetime.h.

References `m_tm`.

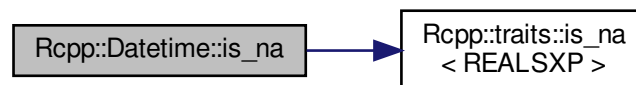
### 6.172.3.13 is\_na()

```
int Rcpp::Datetime::is_na ( ) const [inline]
```

Definition at line 71 of file Datetime.h.

References `Rcpp::traits::is_na< REALSXP >()`, and `m_dt`.

Here is the call graph for this function:



### 6.172.3.14 operator double()

```
Rcpp::Datetime::operator double ( ) const [inline]
```

Definition at line 73 of file Datetime.h.

References `m_dt`.

### 6.172.3.15 update\_tm()

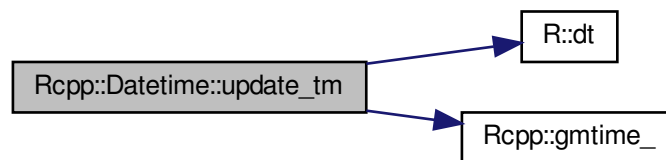
```
void Rcpp::Datetime::update_tm ( ) [inline], [private]
```

Definition at line 100 of file Datetime.h.

References R::dt(), Rcpp::gmtime\_(), m\_dt, m\_tm, and m\_us.

Referenced by Datetime().

Here is the call graph for this function:



## 6.172.4 Friends And Related Function Documentation

### 6.172.4.1 operator"!="

```
bool operator!= (
    const Datetime & dt1,
    const Datetime & dt2 ) [friend]
```

Definition at line 164 of file Datetime.h.

### 6.172.4.2 operator+ [1/2]

```
Datetime operator+ (
    const Datetime & dt,
    double offset ) [friend]
```

Definition at line 138 of file Datetime.h.

### 6.172.4.3 operator+ [2/2]

```
Datetime operator+ (  
    const Datetime & dt,  
    int offset ) [friend]
```

Definition at line 148 of file Datetime.h.

### 6.172.4.4 operator-

```
double operator- (  
    const Datetime & dt1,  
    const Datetime & dt2 ) [friend]
```

Definition at line 158 of file Datetime.h.

### 6.172.4.5 operator<

```
bool operator< (  
    const Datetime & dt1,  
    const Datetime & dt2 ) [friend]
```

Definition at line 159 of file Datetime.h.

### 6.172.4.6 operator<<

```
std::ostream& operator<< (  
    std::ostream & s,  
    const Datetime d ) [friend]
```

Definition at line 166 of file Datetime.h.

### 6.172.4.7 operator<=

```
bool operator<= (  
    const Datetime & dt1,  
    const Datetime & dt2 ) [friend]
```

Definition at line 163 of file Datetime.h.

#### 6.172.4.8 operator==

```
bool operator== (
    const Datetime & dt1,
    const Datetime & dt2 ) [friend]
```

Definition at line 161 of file Datetime.h.

#### 6.172.4.9 operator>

```
bool operator> (
    const Datetime & dt1,
    const Datetime & dt2 ) [friend]
```

Definition at line 160 of file Datetime.h.

#### 6.172.4.10 operator>=

```
bool operator>= (
    const Datetime & dt1,
    const Datetime & dt2 ) [friend]
```

Definition at line 162 of file Datetime.h.

### 6.172.5 Member Data Documentation

#### 6.172.5.1 m\_dt

```
double Rcpp::Datetime::m_dt [private]
```

Definition at line 95 of file Datetime.h.

Referenced by Datetime(), format(), getFractionalTimestamp(), is\_na(), operator double(), and update\_tm().

### 6.172.5.2 m\_tm

```
struct tm Rcpp::Datetime::m_tm [private]
```

Definition at line 95 of file Datetime.h.

Referenced by `getDay()`, `getHours()`, `getMinutes()`, `getMonth()`, `getSeconds()`, `getWeekday()`, `getYear()`, `getYearday()`, and `update_tm()`.

### 6.172.5.3 m\_us

```
unsigned int Rcpp::Datetime::m_us [private]
```

Definition at line 97 of file Datetime.h.

Referenced by `format()`, `getMicroSeconds()`, and `update_tm()`.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/date\\_datetime/Datetime.h](#)

## 6.173 Rcpp::algorithm::helpers::decays\_to\_ctype< T > Struct Template Reference

```
#include <algorithm.h>
```

### Static Public Member Functions

- static [CTYPE\\_CHAR test](#) (const char &)
- static [CTYPE\\_SHORT test](#) (const short &)
- static [CTYPE\\_INT test](#) (const int &)
- static [CTYPE\\_LONG test](#) (const long &)
- static [CTYPE\\_FLOAT test](#) (const float &)
- static [CTYPE\\_DOUBLE test](#) (const double &)
- static [CTYPE\\_LONG\\_DOUBLE test](#) (const long double &)
- static [CTYPE\\_STRING test](#) (const std::string &)
- static [CTYPE\\_UNSIGNED\\_CHAR test](#) (const unsigned char &)
- static [CTYPE\\_UNSIGNED\\_SHORT test](#) (const unsigned short &)
- static [CTYPE\\_UNSIGNED\\_INT test](#) (const unsigned int &)
- static [CTYPE\\_UNSIGNED\\_LONG test](#) (const unsigned long &)
- static [CTYPE\\_UNKNOWN test](#) (...)
- static T [make](#) ()



## Static Public Attributes

- static const bool `value` = `ctype_helper< sizeof(test(make())) >::value`

### 6.173.1 Detailed Description

```
template<typename T>
struct Rcpp::algorithm::helpers::decays_to_ctype< T >
```

Definition at line 136 of file `algorithm.h`.

### 6.173.2 Member Function Documentation

#### 6.173.2.1 `make()`

```
template<typename T >
static T Rcpp::algorithm::helpers::decays_to_ctype< T >::make ( ) [static]
```

#### 6.173.2.2 `test()` [1/13]

```
template<typename T >
static CTYPE_UNKNOWN Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
    ... ) [static]
```

#### 6.173.2.3 `test()` [2/13]

```
template<typename T >
static CTYPE_CHAR Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
    const char & ) [static]
```

#### 6.173.2.4 `test()` [3/13]

```
template<typename T >
static CTYPE_DOUBLE Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
    const double & ) [static]
```

**6.173.2.5 test()** [4/13]

```
template<typename T >
static CTYPE_FLOAT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
    const float & ) [static]
```

**6.173.2.6 test()** [5/13]

```
template<typename T >
static CTYPE_INT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
    const int & ) [static]
```

**6.173.2.7 test()** [6/13]

```
template<typename T >
static CTYPE_LONG Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
    const long & ) [static]
```

**6.173.2.8 test()** [7/13]

```
template<typename T >
static CTYPE_LONG_DOUBLE Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
    const long double & ) [static]
```

**6.173.2.9 test()** [8/13]

```
template<typename T >
static CTYPE_SHORT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
    const short & ) [static]
```

**6.173.2.10 test()** [9/13]

```
template<typename T >
static CTYPE_STRING Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
    const std::string & ) [static]
```

### 6.173.2.11 test() [10/13]

```
template<typename T >
static CTYPE_UNSIGNED_CHAR Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
    const unsigned char & ) [static]
```

### 6.173.2.12 test() [11/13]

```
template<typename T >
static CTYPE_UNSIGNED_INT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
    const unsigned int & ) [static]
```

### 6.173.2.13 test() [12/13]

```
template<typename T >
static CTYPE_UNSIGNED_LONG Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
    const unsigned long & ) [static]
```

### 6.173.2.14 test() [13/13]

```
template<typename T >
static CTYPE_UNSIGNED_SHORT Rcpp::algorithm::helpers::decays_to_ctype< T >::test (
    const unsigned short & ) [static]
```

## 6.173.3 Member Data Documentation

### 6.173.3.1 value

```
template<typename T >
const bool Rcpp::algorithm::helpers::decays_to_ctype< T >::value = ctype_helper< sizeof(test(make()))
>::value [static]
```

Definition at line 160 of file algorithm.h.

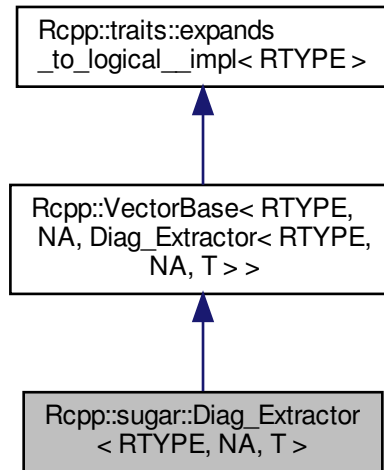
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

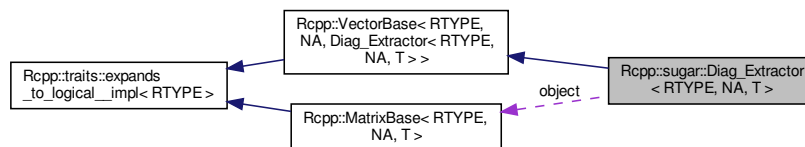
## 6.174 Rcpp::sugar::Diag\_Extractor< RTYPE, NA, T > Class Template Reference

```
#include <diag.h>
```

Inheritance diagram for Rcpp::sugar::Diag\_Extractor< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Diag\_Extractor< RTYPE, NA, T >:



### Public Types

- typedef [Rcpp::MatrixBase< RTYPE, NA, T > MAT\\_TYPE](#)
- typedef [Rcpp::traits::storage\\_type< RTYPE >::type STORAGE](#)

## Public Member Functions

- [Diag\\_Extractor](#) (const [MAT\\_TYPE](#) &object\_)
- [STORAGE operator\[\]](#) (int i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [MAT\\_TYPE](#) & [object](#)
- [R\\_xlen\\_t n](#)

### 6.174.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >
```

Definition at line 29 of file diag.h.

### 6.174.2 Member Typedef Documentation

#### 6.174.2.1 MAT\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::MatrixBase<RTYPE,NA,T> Rcpp::sugar::Diag\_Extractor< RTYPE, NA, T >::MAT\_TYPE
```

Definition at line 31 of file diag.h.

#### 6.174.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Diag\_Extractor< RTYPE, NA, T >↔
::STORAGE
```

Definition at line 32 of file diag.h.

### 6.174.3 Constructor & Destructor Documentation

### 6.174.3.1 Diag\_Extractor()

```
template<int RTYPE, bool NA, typename T >  
Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::Diag_Extractor (  
    const MAT_TYPE & object_ ) [inline]
```

Definition at line 34 of file diag.h.

References Rcpp::sugar::Diag\_Extractor< RTYPE, NA, T >::n.

## 6.174.4 Member Function Documentation

### 6.174.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >  
STORAGE Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::operator[] (  
    int i ) const [inline]
```

Definition at line 40 of file diag.h.

References Rcpp::sugar::Diag\_Extractor< RTYPE, NA, T >::object.

### 6.174.4.2 size()

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 43 of file diag.h.

References Rcpp::sugar::Diag\_Extractor< RTYPE, NA, T >::n.

## 6.174.5 Member Data Documentation

### 6.174.5.1 n

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::n [private]
```

Definition at line 47 of file diag.h.

Referenced by Rcpp::sugar::Diag\_Extractor< RTYPE, NA, T >::Diag\_Extractor(), and Rcpp::sugar::Diag\_Extractor< RTYPE, NA, T >::size().

## 6.174.5.2 object

```
template<int RTYPE, bool NA, typename T >
const MAT_TYPE& Rcpp::sugar::Diag_Extractor< RTYPE, NA, T >::object [private]
```

Definition at line 46 of file diag.h.

Referenced by Rcpp::sugar::Diag\_Extractor< RTYPE, NA, T >::operator[]().

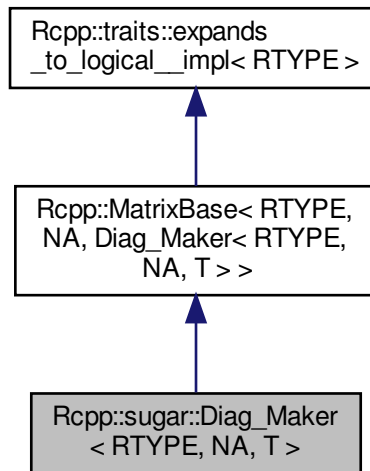
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/diag.h

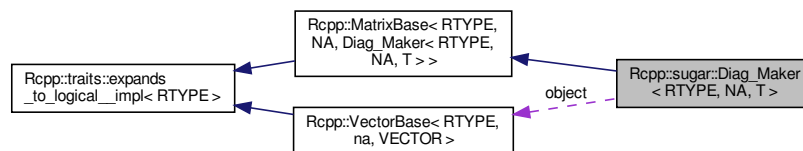
## 6.175 Rcpp::sugar::Diag\_Maker< RTYPE, NA, T > Class Template Reference

```
#include <diag.h>
```

Inheritance diagram for Rcpp::sugar::Diag\_Maker< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Diag\_Maker< RTYPE, NA, T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, NA, T > [VEC\\_TYPE](#)
- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)

## Public Member Functions

- [Diag\\_Maker](#) (const [VEC\\_TYPE](#) &object\_)
- [STORAGE operator\(\)](#) (int i, int j) const
- [R\\_xlen\\_t size](#) () const
- int [ncol](#) () const
- int [nrow](#) () const

## Private Attributes

- const [VEC\\_TYPE](#) & [object](#)
- int [n](#)

## Additional Inherited Members

### 6.175.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Diag_Maker< RTYPE, NA, T >
```

Definition at line 52 of file diag.h.

### 6.175.2 Member Typedef Documentation

#### 6.175.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Diag\_Maker< RTYPE, NA, T >::STORAGE
```

Definition at line 55 of file diag.h.



### 6.175.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 54 of file diag.h.

## 6.175.3 Constructor & Destructor Documentation

### 6.175.3.1 Diag\_Maker()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::Diag_Maker (
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 57 of file diag.h.

## 6.175.4 Member Function Documentation

### 6.175.4.1 ncol()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::ncol ( ) const [inline]
```

Definition at line 63 of file diag.h.

References Rcpp::sugar::Diag\_Maker< RTYPE, NA, T >::n.

### 6.175.4.2 nrow()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::nrow ( ) const [inline]
```

Definition at line 64 of file diag.h.

References Rcpp::sugar::Diag\_Maker< RTYPE, NA, T >::n.

### 6.175.4.3 operator()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::operator() (
    int i,
    int j ) const [inline]
```

Definition at line 59 of file diag.h.

### 6.175.4.4 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 62 of file diag.h.

References Rcpp::sugar::Diag\_Maker< RTYPE, NA, T >::n.

## 6.175.5 Member Data Documentation

### 6.175.5.1 n

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::n [private]
```

Definition at line 68 of file diag.h.

Referenced by Rcpp::sugar::Diag\_Maker< RTYPE, NA, T >::ncol(), Rcpp::sugar::Diag\_Maker< RTYPE, NA, T >::nrow(), and Rcpp::sugar::Diag\_Maker< RTYPE, NA, T >::size().

### 6.175.5.2 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Diag_Maker< RTYPE, NA, T >::object [private]
```

Definition at line 67 of file diag.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/diag.h

## 6.176 Rcpp::sugar::diag\_result\_type\_trait< T > Struct Template Reference

```
#include <diag.h>
```

### Public Types

- typedef [Rcpp::traits::if\\_< Rcpp::traits::matrix\\_interface< T >::value, Diag\\_Extractor< T::r\\_type::value, T::can\\_have\\_na::value, T >, Diag\\_Maker< T::r\\_type::value, T::can\\_have\\_na::value, T > >::type](#) type

### 6.176.1 Detailed Description

```
template<typename T>  
struct Rcpp::sugar::diag_result_type_trait< T >
```

Definition at line 71 of file diag.h.

### 6.176.2 Member Typedef Documentation

#### 6.176.2.1 type

```
template<typename T >  
typedef Rcpp::traits::if\_< Rcpp::traits::matrix\_interface<T>::value, Diag\_Extractor< T::r\_type::value, T::can\_have\_na::value, T >, Diag\_Maker< T::r\_type::value, T::can\_have\_na::value, T > >::type Rcpp::sugar::diag_result_type_trait< T >::type
```

Definition at line 76 of file diag.h.

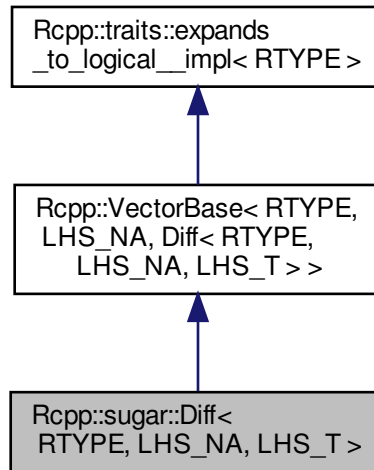
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/matrix/diag.h](#)

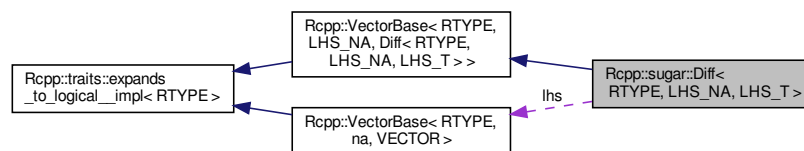
## 6.177 Rcpp::sugar::Diff< RTYPE, LHS\_NA, LHS\_T > Class Template Reference

```
#include <diff.h>
```

Inheritance diagram for Rcpp::sugar::Diff< RTYPE, LHS\_NA, LHS\_T >:



Collaboration diagram for Rcpp::sugar::Diff< RTYPE, LHS\_NA, LHS\_T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T >` `LHS_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`

## Public Member Functions

- [Diff](#) (const [LHS\\_TYPE](#) &lhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- void [set\\_previous](#) (R\_xlen\_t i, [STORAGE](#) value) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_TYPE](#) & lhs
- [STORAGE](#) previous
- R\_xlen\_t [previous\\_index](#)
- bool [was\\_na](#)

### 6.177.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >
```

Definition at line 32 of file diff.h.

### 6.177.2 Member Typedef Documentation

#### 6.177.2.1 LHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef Rcpp::VectorBase<RTYPE, LHS_NA, LHS_T> Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::LHS_TYPE
```

Definition at line 34 of file diff.h.

#### 6.177.2.2 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >↵
::STORAGE
```

Definition at line 35 of file diff.h.

## 6.177.3 Constructor & Destructor Documentation

### 6.177.3.1 Diff()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::Diff (
    const LHS_TYPE & lhs_ ) [inline]
```

Definition at line 37 of file diff.h.

## 6.177.4 Member Function Documentation

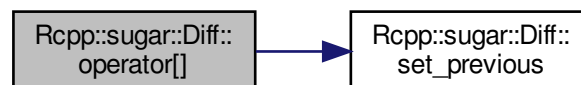
### 6.177.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
STORAGE Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 44 of file diff.h.

References `Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::lhs`, `Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous`, `Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous_index`, `Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::set_previous()`, and `Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::was_na`.

Here is the call graph for this function:



### 6.177.4.2 set\_previous()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
void Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::set_previous (
    R_xlen_t i,
    STORAGE value ) const [inline]
```

Definition at line 59 of file diff.h.

References Rcpp::sugar::Diff< RTYPE, LHS\_NA, LHS\_T >::previous, Rcpp::sugar::Diff< RTYPE, LHS\_NA, LHS\_T >::previous\_index, and Rcpp::sugar::Diff< RTYPE, LHS\_NA, LHS\_T >::was\_na.

Referenced by Rcpp::sugar::Diff< RTYPE, LHS\_NA, LHS\_T >::operator[]().

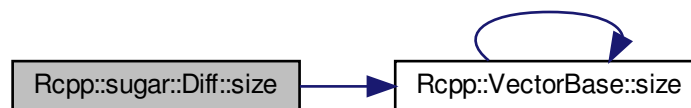
### 6.177.4.3 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::size ( ) const [inline]
```

Definition at line 65 of file diff.h.

References Rcpp::sugar::Diff< RTYPE, LHS\_NA, LHS\_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



## 6.177.5 Member Data Documentation

### 6.177.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
const LHS_TYPE& Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::lhs [private]
```

Definition at line 68 of file diff.h.

Referenced by `Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::operator[]()`, `Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::operator[]()`, `Rcpp::sugar::Diff< RTYPE, false, LHS_T >::operator[]()`, `Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::size()`, `Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::size()`, and `Rcpp::sugar::Diff< RTYPE, false, LHS_T >::size()`.

### 6.177.5.2 previous

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
STORAGE Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous [mutable], [private]
```

Definition at line 69 of file diff.h.

Referenced by `Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::operator[]()`, `Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::operator[]()`, `Rcpp::sugar::Diff< RTYPE, false, LHS_T >::operator[]()`, and `Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::set_previous()`.

### 6.177.5.3 previous\_index

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::previous_index [mutable], [private]
```

Definition at line 70 of file diff.h.

Referenced by `Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::operator[]()`, `Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::operator[]()`, `Rcpp::sugar::Diff< RTYPE, false, LHS_T >::operator[]()`, and `Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::set_previous()`.

### 6.177.5.4 was\_na

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
bool Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::was_na [mutable], [private]
```

Definition at line 71 of file diff.h.

Referenced by `Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::operator[]()`, and `Rcpp::sugar::Diff< RTYPE, LHS_NA, LHS_T >::set_previous()`.

The documentation for this class was generated from the following file:

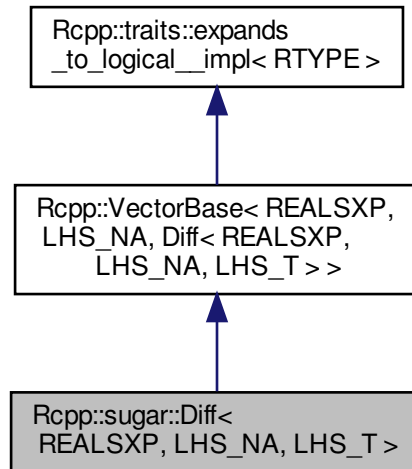
- `inst/include/Rcpp/sugar/functions/diff.h`



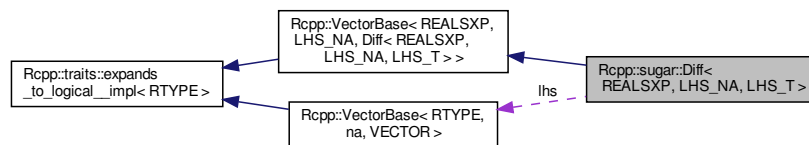
## 6.178 Rcpp::sugar::Diff< REALSXP, LHS\_NA, LHS\_T > Class Template Reference

```
#include <diff.h>
```

Inheritance diagram for Rcpp::sugar::Diff< REALSXP, LHS\_NA, LHS\_T >:



Collaboration diagram for Rcpp::sugar::Diff< REALSXP, LHS\_NA, LHS\_T >:



### Public Types

- typedef `Rcpp::VectorBase< REALSXP, LHS_NA, LHS_T >` `LHS_TYPE`

### Public Member Functions

- `Diff` (const `LHS_TYPE` &lhs\_)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const [LHS\\_TYPE](#) & *lhs*
- double *previous*
- [R\\_xlen\\_t](#) *previous\_index*

### 6.178.1 Detailed Description

```
template<typename LHS_T, bool LHS_NA>  
class Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >
```

Definition at line 75 of file diff.h.

### 6.178.2 Member Typedef Documentation

#### 6.178.2.1 LHS\_TYPE

```
template<typename LHS_T , bool LHS_NA>  
typedef Rcpp::VectorBase<REALSXP, LHS_NA, LHS_T> Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >↔  
::LHS_TYPE
```

Definition at line 77 of file diff.h.

### 6.178.3 Constructor & Destructor Documentation

#### 6.178.3.1 Diff()

```
template<typename LHS_T , bool LHS_NA>  
Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::Diff (  
    const LHS\_TYPE & lhs_ ) [inline]
```

Definition at line 79 of file diff.h.

### 6.178.4 Member Function Documentation

### 6.178.4.1 operator[]()

```
template<typename LHS_T , bool LHS_NA>
double Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 81 of file diff.h.

References [Rcpp::sugar::Diff< RTYPE, LHS\\_NA, LHS\\_T >::lhs](#), [Rcpp::sugar::Diff< RTYPE, LHS\\_NA, LHS\\_T >::previous](#), and [Rcpp::sugar::Diff< RTYPE, LHS\\_NA, LHS\\_T >::previous\\_index](#).

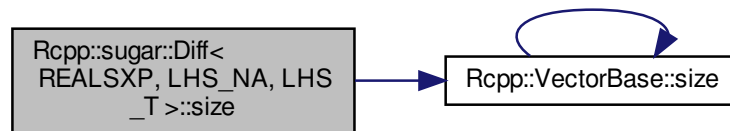
### 6.178.4.2 size()

```
template<typename LHS_T , bool LHS_NA>
R_xlen_t Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::size ( ) const [inline]
```

Definition at line 89 of file diff.h.

References [Rcpp::sugar::Diff< RTYPE, LHS\\_NA, LHS\\_T >::lhs](#), and [Rcpp::VectorBase< RTYPE, na, VECTOR >::size\(\)](#).

Here is the call graph for this function:



## 6.178.5 Member Data Documentation

### 6.178.5.1 lhs

```
template<typename LHS_T , bool LHS_NA>
const LHS_TYPE& Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::lhs [private]
```

Definition at line 92 of file diff.h.

### 6.178.5.2 previous

```
template<typename LHS_T , bool LHS_NA>
double Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::previous [mutable], [private]
```

Definition at line 93 of file diff.h.

### 6.178.5.3 previous\_index

```
template<typename LHS_T , bool LHS_NA>
R_xlen_t Rcpp::sugar::Diff< REALSXP, LHS_NA, LHS_T >::previous_index [mutable], [private]
```

Definition at line 94 of file diff.h.

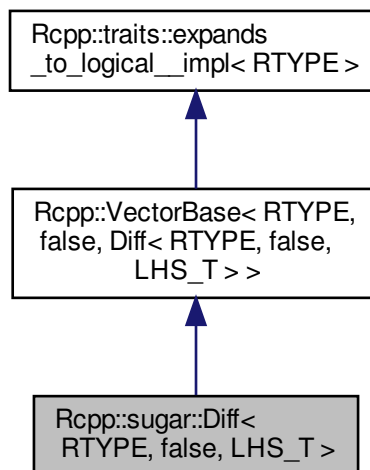
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/diff.h](#)

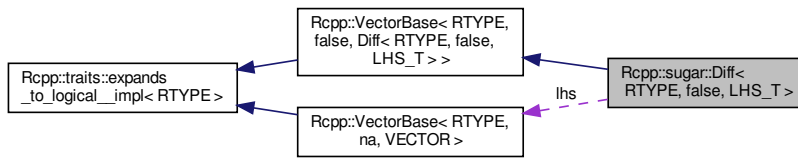
## 6.179 Rcpp::sugar::Diff< RTYPE, false, LHS\_T > Class Template Reference

```
#include <diff.h>
```

Inheritance diagram for Rcpp::sugar::Diff< RTYPE, false, LHS\_T >:



Collaboration diagram for Rcpp::sugar::Diff< RTYPE, false, LHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase< RTYPE, false, LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) [STORAGE](#)

## Public Member Functions

- [Diff](#) (const [LHS\\_TYPE](#) &lhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [LHS\\_TYPE](#) &lhs
- [STORAGE](#) previous
- R\_xlen\_t [previous\\_index](#)

### 6.179.1 Detailed Description

```

template<int RTYPE, typename LHS_T>
class Rcpp::sugar::Diff< RTYPE, false, LHS_T >
  
```

Definition at line 98 of file diff.h.

### 6.179.2 Member Typedef Documentation

### 6.179.2.1 LHS\_TYPE

```
template<int RTYPE, typename LHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Diff< RTYPE, false, LHS_T >::LHS_TYPE
```

Definition at line 100 of file diff.h.

### 6.179.2.2 STORAGE

```
template<int RTYPE, typename LHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Diff< RTYPE, false, LHS_T >::STORAGE
```

Definition at line 101 of file diff.h.

## 6.179.3 Constructor & Destructor Documentation

### 6.179.3.1 Diff()

```
template<int RTYPE, typename LHS_T >
Rcpp::sugar::Diff< RTYPE, false, LHS_T >::Diff (
    const LHS_TYPE & lhs_ ) [inline]
```

Definition at line 103 of file diff.h.

## 6.179.4 Member Function Documentation

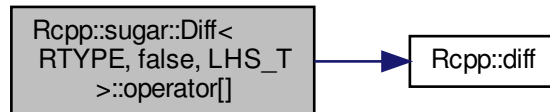
### 6.179.4.1 operator[]()

```
template<int RTYPE, typename LHS_T >
STORAGE Rcpp::sugar::Diff< RTYPE, false, LHS_T >::operator[] (
    R_xlen_t i) const [inline]
```

Definition at line 105 of file diff.h.

References [Rcpp::diff\(\)](#), [Rcpp::sugar::Diff< RTYPE, LHS\\_NA, LHS\\_T >::lhs](#), [Rcpp::sugar::Diff< RTYPE, LHS\\_NA, LHS\\_T >::previous](#), and [Rcpp::sugar::Diff< RTYPE, LHS\\_NA, LHS\\_T >::previous\\_index](#).

Here is the call graph for this function:



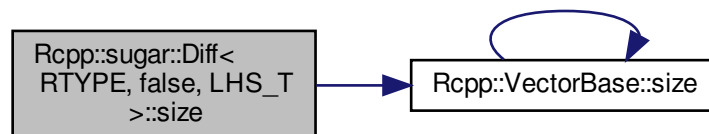
### 6.179.4.2 size()

```
template<int RTYPE, typename LHS_T >
R_xlen_t Rcpp::sugar::Diff< RTYPE, false, LHS_T >::size ( ) const [inline]
```

Definition at line 113 of file diff.h.

References [Rcpp::sugar::Diff< RTYPE, LHS\\_NA, LHS\\_T >::lhs](#), and [Rcpp::VectorBase< RTYPE, na, VECTOR >::size\(\)](#).

Here is the call graph for this function:



## 6.179.5 Member Data Documentation

### 6.179.5.1 lhs

```
template<int RTYPE, typename LHS_T >
const LHS_TYPE& Rcpp::sugar::Diff< RTYPE, false, LHS_T >::lhs [private]
```

Definition at line 116 of file diff.h.

### 6.179.5.2 previous

```
template<int RTYPE, typename LHS_T >
STORAGE Rcpp::sugar::Diff< RTYPE, false, LHS_T >::previous [mutable], [private]
```

Definition at line 117 of file diff.h.

### 6.179.5.3 previous\_index

```
template<int RTYPE, typename LHS_T >
R_xlen_t Rcpp::sugar::Diff< RTYPE, false, LHS_T >::previous_index [mutable], [private]
```

Definition at line 118 of file diff.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/diff.h](#)

## 6.180 Rcpp::Dimension Class Reference

```
#include <Dimension.h>
```

### Public Types

- typedef std::vector< int >::reference reference
- typedef std::vector< int >::const\_reference const\_reference



## Public Member Functions

- [Dimension](#) ()
- [Dimension](#) (SEXP [dims](#))
- [Dimension](#) (const [Dimension](#) &other)
- [Dimension](#) & [operator=](#) (const [Dimension](#) &other)
- [Dimension](#) (const [size\\_t](#) &n1)
- [Dimension](#) (const [size\\_t](#) &n1, const [size\\_t](#) &n2)
- [Dimension](#) (const [size\\_t](#) &n1, const [size\\_t](#) &n2, const [size\\_t](#) &n3)
- [operator SEXP](#) () const
- [int](#) [size](#) () const
- [R\\_xlen\\_t](#) [prod](#) () const
- [reference operator\[\]](#) (int i)
- [const\\_reference operator\[\]](#) (int i) const

## Private Attributes

- `std::vector< int >` [dims](#)

### 6.180.1 Detailed Description

Definition at line 27 of file [Dimension.h](#).

### 6.180.2 Member Typedef Documentation

#### 6.180.2.1 `const_reference`

```
typedef std::vector<int>::const_reference Rcpp::Dimension::const_reference
```

Definition at line 30 of file [Dimension.h](#).

#### 6.180.2.2 `reference`

```
typedef std::vector<int>::reference Rcpp::Dimension::reference
```

Definition at line 29 of file [Dimension.h](#).

### 6.180.3 Constructor & Destructor Documentation

**6.180.3.1 Dimension()** [1/6]

```
Rcpp::Dimension::Dimension ( ) [inline]
```

Definition at line 32 of file Dimension.h.

**6.180.3.2 Dimension()** [2/6]

```
Rcpp::Dimension::Dimension (
    SEXP dims ) [inline]
```

Definition at line 23 of file Dimension.h.

**6.180.3.3 Dimension()** [3/6]

```
Rcpp::Dimension::Dimension (
    const Dimension & other ) [inline]
```

Definition at line 36 of file Dimension.h.

**6.180.3.4 Dimension()** [4/6]

```
Rcpp::Dimension::Dimension (
    const size_t & n1 ) [inline]
```

Definition at line 42 of file Dimension.h.

References [dims](#).

**6.180.3.5 Dimension()** [5/6]

```
Rcpp::Dimension::Dimension (
    const size_t & n1,
    const size_t & n2 ) [inline]
```

Definition at line 45 of file Dimension.h.

References [dims](#).

### 6.180.3.6 Dimension() [6/6]

```
Rcpp::Dimension::Dimension (
    const size_t & n1,
    const size_t & n2,
    const size_t & n3 ) [inline]
```

Definition at line 49 of file Dimension.h.

References dims.

## 6.180.4 Member Function Documentation

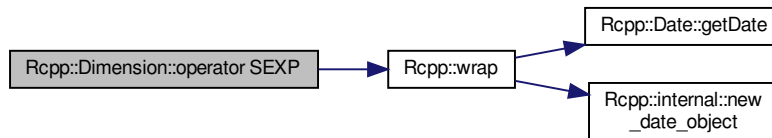
### 6.180.4.1 operator SEXP()

```
Rcpp::Dimension::operator SEXP ( ) const [inline]
```

Definition at line 25 of file Dimension.h.

References Rcpp::wrap().

Here is the call graph for this function:



### 6.180.4.2 operator=()

```
Dimension& Rcpp::Dimension::operator= (
    const Dimension & other ) [inline]
```

Definition at line 37 of file Dimension.h.

References dims.

### 6.180.4.3 operator[]() [1/2]

```
reference Rcpp::Dimension::operator[] (
    int i ) [inline]
```

Definition at line 63 of file Dimension.h.

References dims.

### 6.180.4.4 operator[]() [2/2]

```
const_reference Rcpp::Dimension::operator[] (
    int i ) const [inline]
```

Definition at line 67 of file Dimension.h.

References dims.

### 6.180.4.5 prod()

```
R_xlen_t Rcpp::Dimension::prod ( ) const [inline]
```

Definition at line 59 of file Dimension.h.

References dims.

### 6.180.4.6 size()

```
int Rcpp::Dimension::size ( ) const [inline]
```

Definition at line 56 of file Dimension.h.

References dims.

## 6.180.5 Member Data Documentation

### 6.180.5.1 dims

```
std::vector<int> Rcpp::Dimension::dims [private]
```

Definition at line 73 of file Dimension.h.

Referenced by Dimension(), operator=(), operator[](), prod(), and size().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Dimension.h

## 6.181 Rcpp::internal::DimNameProxy Class Reference

```
#include <DimNameProxy.h>
```

### Public Member Functions

- [DimNameProxy](#) (SEXP data, int dim)
- [DimNameProxy](#) ([DimNameProxy](#) const &other)
- [DimNameProxy](#) & [assign](#) (SEXP other)
- [DimNameProxy](#) & [operator=](#) (SEXP other)
- [DimNameProxy](#) & [operator=](#) (const [DimNameProxy](#) &other)
- [operator SEXP](#) () const
- [template<typename T >](#)  
[operator T](#) () const

### Private Attributes

- SEXP [data\\_](#)
- int [dim\\_](#)

### 6.181.1 Detailed Description

Definition at line 28 of file DimNameProxy.h.

### 6.181.2 Constructor & Destructor Documentation

### 6.181.2.1 DimNameProxy() [1/2]

```
Rcpp::internal::DimNameProxy::DimNameProxy (
    SEXP data,
    int dim ) [inline]
```

Definition at line 32 of file DimNameProxy.h.

### 6.181.2.2 DimNameProxy() [2/2]

```
Rcpp::internal::DimNameProxy::DimNameProxy (
    DimNameProxy const & other ) [inline]
```

Definition at line 33 of file DimNameProxy.h.

## 6.181.3 Member Function Documentation

### 6.181.3.1 assign()

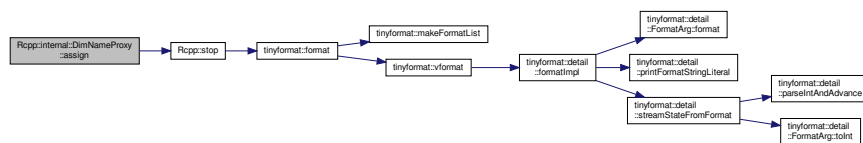
```
DimNameProxy& Rcpp::internal::DimNameProxy::assign (
    SEXP other ) [inline]
```

Definition at line 36 of file DimNameProxy.h.

References `data_`, `dim_`, and `Rcpp::stop()`.

Referenced by operator=().

Here is the call graph for this function:



### 6.181.3.2 operator SEXP()

```
Rcpp::internal::DimNameProxy::operator SEXP ( ) const [inline]
```

Definition at line 66 of file DimNameProxy.h.

References `data_`, and `dim_`.

### 6.181.3.3 operator T()

```
template<typename T >
Rcpp::internal::DimNameProxy::operator T ( ) const [inline]
```

Definition at line 72 of file DimNameProxy.h.

References `data_`, and `dim_`.

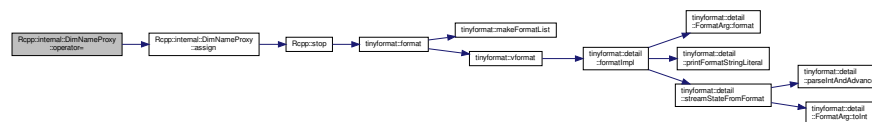
### 6.181.3.4 operator=() [1/2]

```
DimNameProxy& Rcpp::internal::DimNameProxy::operator= (
    const DimNameProxy & other ) [inline]
```

Definition at line 62 of file DimNameProxy.h.

References `assign()`.

Here is the call graph for this function:



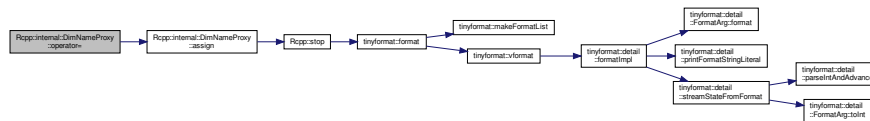
### 6.181.3.5 operator=() [2/2]

```
DimNameProxy& Rcpp::internal::DimNameProxy::operator= (
    SEXP other ) [inline]
```

Definition at line 58 of file DimNameProxy.h.

References `assign()`.

Here is the call graph for this function:



## 6.181.4 Member Data Documentation

### 6.181.4.1 data\_

```
SEXP Rcpp::internal::DimNameProxy::data_ [private]
```

Definition at line 83 of file DimNameProxy.h.

Referenced by `assign()`, `operator SEXP()`, and `operator T()`.

### 6.181.4.2 dim\_

```
int Rcpp::internal::DimNameProxy::dim_ [private]
```

Definition at line 84 of file DimNameProxy.h.

Referenced by `assign()`, `operator SEXP()`, and `operator T()`.

The documentation for this class was generated from the following file:

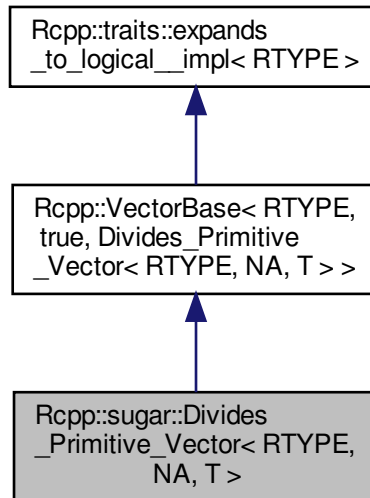
- `inst/include/Rcpp/vector/DimNameProxy.h`



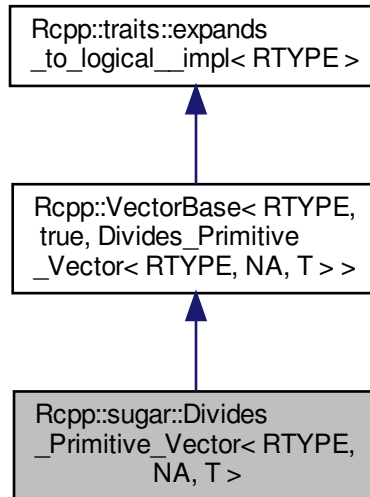
## 6.182 Rcpp::sugar::Divides\_Primitive\_Vector< RTYPE, NA, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Primitive\_Vector< RTYPE, NA, T >:



Collaboration diagram for `Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >`:



## Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< RTYPE, NA, T >::type` `VEC_EXT`
- typedef `traits::storage_type< RTYPE >::type` `STORAGE`

## Public Member Functions

- `Divides_Primitive_Vector (STORAGE lhs_, const VEC_TYPE &rhs_)`
- `STORAGE operator[] (R_xlen_t i) const`
- `R_xlen_t size () const`

## Private Attributes

- `STORAGE lhs`
- `const VEC_EXT & rhs`
- `bool lhs_na`

### 6.182.1 Detailed Description

```

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >
  
```

Definition at line 322 of file `divides.h`.

## 6.182.2 Member Typedef Documentation

### 6.182.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T
>::STORAGE
```

Definition at line 327 of file divides.h.

### 6.182.2.2 VEC\_EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor<RTYPE,NA,T>::type Rcpp::sugar::Divides_Primitive_Vector< RTYPE,
NA, T >::VEC_EXT
```

Definition at line 326 of file divides.h.

### 6.182.2.3 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >←
::VEC_TYPE
```

Definition at line 325 of file divides.h.

## 6.182.3 Constructor & Destructor Documentation

### 6.182.3.1 Divides\_Primitive\_Vector()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::Divides_Primitive_Vector (
    STORAGE lhs_,
    const VEC_TYPE & rhs_ ) [inline]
```

Definition at line 329 of file divides.h.

## 6.182.4 Member Function Documentation

### 6.182.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 332 of file divides.h.

References [Rcpp::sugar::Divides\\_Primitive\\_Vector< RTYPE, NA, T >::lhs](#), [Rcpp::sugar::Divides\\_Primitive\\_Vector< RTYPE, NA, T >::lhs\\_na](#), and [Rcpp::sugar::Divides\\_Primitive\\_Vector< RTYPE, NA, T >::rhs](#).

### 6.182.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 337 of file divides.h.

References [Rcpp::sugar::Divides\\_Primitive\\_Vector< RTYPE, NA, T >::rhs](#).

## 6.182.5 Member Data Documentation

### 6.182.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::lhs [private]
```

Definition at line 339 of file divides.h.

Referenced by [Rcpp::sugar::Divides\\_Primitive\\_Vector< RTYPE, NA, T >::operator\[\]\(\)](#), [Rcpp::sugar::Divides\\_Primitive\\_Vector< REALSXP, NA, T >::operator\[\]\(\)](#), [Rcpp::sugar::Divides\\_Primitive\\_Vector< RTYPE, false, T >::operator\[\]\(\)](#), and [Rcpp::sugar::Divides\\_Primitive\\_Vector< REALSXP, false, T >::operator\[\]\(\)](#).

### 6.182.5.2 lhs\_na

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::lhs_na [private]
```

Definition at line 341 of file divides.h.

Referenced by Rcpp::sugar::Divides\_Primitive\_Vector< RTYPE, NA, T >::operator[](), and Rcpp::sugar::Divides\_← Primitive\_Vector< RTYPE, false, T >::operator[]().

### 6.182.5.3 rhs

```
template<int RTYPE, bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::rhs [private]
```

Definition at line 340 of file divides.h.

Referenced by Rcpp::sugar::Divides\_Primitive\_Vector< RTYPE, NA, T >::operator[](), Rcpp::sugar::Divides\_← Primitive\_Vector< REALSXP, NA, T >::operator[](), Rcpp::sugar::Divides\_Primitive\_Vector< RTYPE, false, T >← ::operator[](), Rcpp::sugar::Divides\_Primitive\_Vector< REALSXP, false, T >::operator[](), Rcpp::sugar::Divides\_← Primitive\_Vector< RTYPE, NA, T >::size(), Rcpp::sugar::Divides\_Primitive\_Vector< REALSXP, NA, T >::size(), Rcpp::sugar::Divides\_Primitive\_Vector< RTYPE, false, T >::size(), and Rcpp::sugar::Divides\_Primitive\_Vector< RE- ALSXP, false, T >::size().

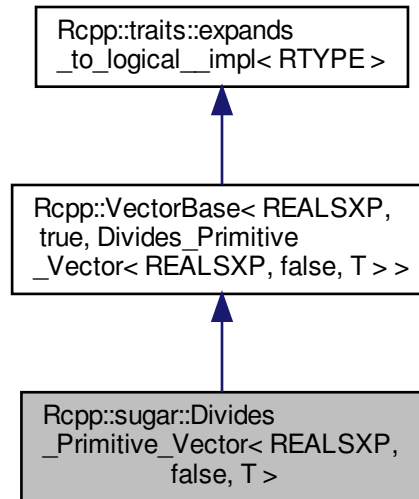
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/divides.h](#)

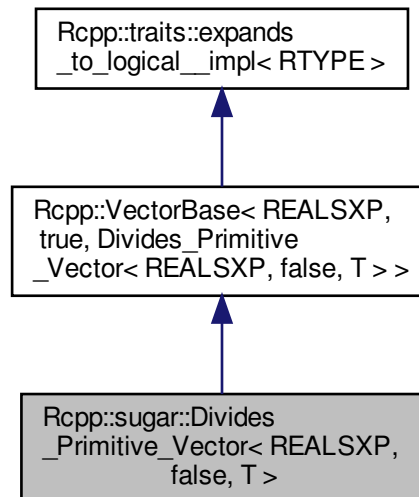
## 6.183 Rcpp::sugar::Divides\_Primitive\_Vector< REALSXP, false, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for `Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >`:



Collaboration diagram for `Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >`:



## Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, false, T > [VEC\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, T >::type [VEC\\_EXT](#)

## Public Member Functions

- [Divides\\_Primitive\\_Vector](#) (double lhs\_, const [VEC\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- double [lhs](#)
- const [VEC\\_EXT](#) & [rhs](#)

### 6.183.1 Detailed Description

```
template<typename T>
class Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >
```

Definition at line 389 of file divides.h.

### 6.183.2 Member Typedef Documentation

#### 6.183.2.1 VEC\_EXT

```
template<typename T >
typedef Rcpp::traits::Extractor<REALSXP,false,T>::type Rcpp::sugar::Divides\_Primitive\_Vector<
REALSXP, false, T >::VEC_EXT
```

Definition at line 393 of file divides.h.

#### 6.183.2.2 VEC\_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP,false,T> Rcpp::sugar::Divides\_Primitive\_Vector< REALSXP, false,
T >::VEC_TYPE
```

Definition at line 392 of file divides.h.

## 6.183.3 Constructor & Destructor Documentation

### 6.183.3.1 Divides\_Primitive\_Vector()

```
template<typename T >
Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::Divides_Primitive_Vector (
    double lhs_,
    const VEC_TYPE & rhs_ ) [inline]
```

Definition at line 395 of file divides.h.

## 6.183.4 Member Function Documentation

### 6.183.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 398 of file divides.h.

References [Rcpp::sugar::Divides\\_Primitive\\_Vector< RTYPE, NA, T >::lhs](#), and [Rcpp::sugar::Divides\\_Primitive\\_Vector< RTYPE, NA, T >::rhs](#).

### 6.183.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::size ( ) const [inline]
```

Definition at line 401 of file divides.h.

References [Rcpp::sugar::Divides\\_Primitive\\_Vector< RTYPE, NA, T >::rhs](#).

## 6.183.5 Member Data Documentation



## 6.183.5.1 lhs

```
template<typename T >
double Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::lhs [private]
```

Definition at line 404 of file divides.h.

## 6.183.5.2 rhs

```
template<typename T >
const VEC_EXT& Rcpp::sugar::Divides_Primitive_Vector< REALSXP, false, T >::rhs [private]
```

Definition at line 405 of file divides.h.

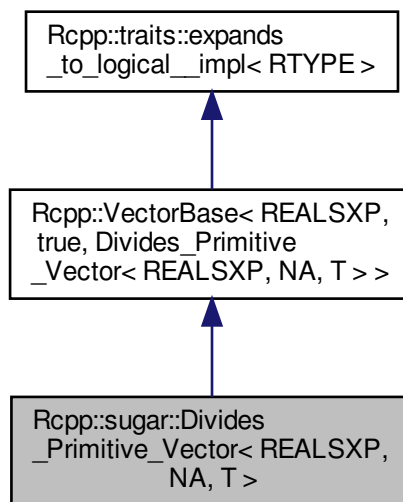
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/divides.h](#)

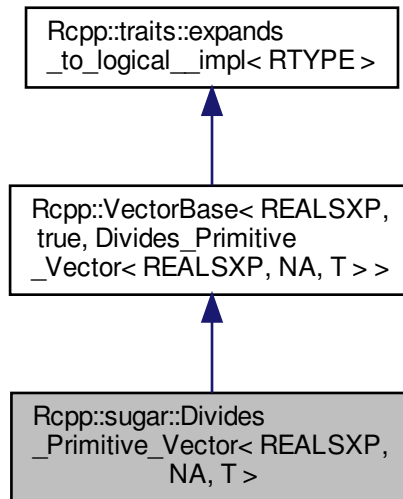
## 6.184 Rcpp::sugar::Divides\_Primitive\_Vector< REALSXP, NA, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Primitive\_Vector< REALSXP, NA, T >:



Collaboration diagram for `Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >`:



## Public Types

- typedef `Rcpp::VectorBase< REALSXP, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< REALSXP, NA, T >::type` `VEC_EXT`

## Public Member Functions

- `Divides_Primitive_Vector` (double lhs\_, const `VEC_TYPE` &rhs\_)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- double lhs
- const `VEC_EXT` & rhs

### 6.184.1 Detailed Description

```

template<bool NA, typename T>
class Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >
  
```

Definition at line 345 of file `divides.h`.

## 6.184.2 Member Typedef Documentation

### 6.184.2.1 VEC\_EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP,NA,T>::type Rcpp::sugar::Divides_Primitive_Vector< REALSXP,
NA, T >::VEC_EXT
```

Definition at line 349 of file divides.h.

### 6.184.2.2 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >←
::VEC_TYPE
```

Definition at line 348 of file divides.h.

## 6.184.3 Constructor & Destructor Documentation

### 6.184.3.1 Divides\_Primitive\_Vector()

```
template<bool NA, typename T >
Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >::Divides_Primitive_Vector (
    double lhs_,
    const VEC_TYPE & rhs_ ) [inline]
```

Definition at line 351 of file divides.h.

## 6.184.4 Member Function Documentation

#### 6.184.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >::operator[] (
    R_xlen_t i) const [inline]
```

Definition at line 354 of file divides.h.

References `Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::lhs`, and `Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::rhs`.

#### 6.184.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >::size ( ) const [inline]
```

Definition at line 357 of file divides.h.

References `Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::rhs`.

### 6.184.5 Member Data Documentation

#### 6.184.5.1 lhs

```
template<bool NA, typename T >
double Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >::lhs [private]
```

Definition at line 359 of file divides.h.

#### 6.184.5.2 rhs

```
template<bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Divides_Primitive_Vector< REALSXP, NA, T >::rhs [private]
```

Definition at line 360 of file divides.h.

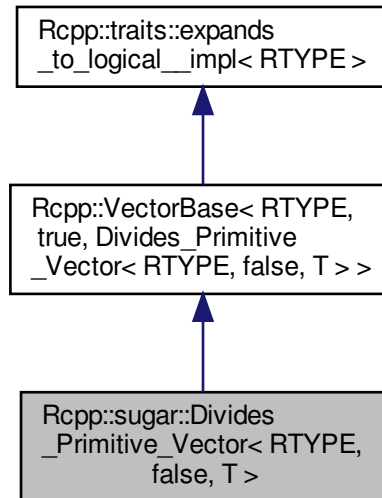
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/operators/divides.h`

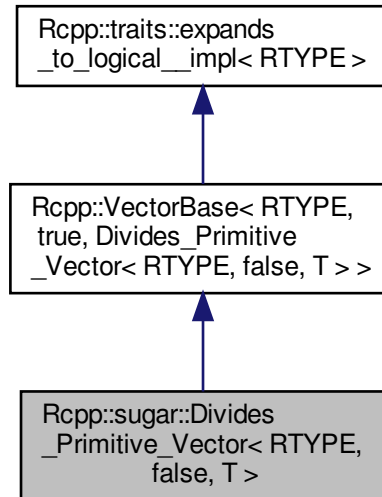
## 6.185 Rcpp::sugar::Divides\_Primitive\_Vector< RTYPE, false, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Primitive\_Vector< RTYPE, false, T >:



Collaboration diagram for `Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >`:



## Public Types

- typedef `Rcpp::VectorBase< RTYPE, false, T >` `VEC_TYPE`
- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::traits::Extractor< RTYPE, false, T >::type` `VEC_EXT`

## Public Member Functions

- `Divides_Primitive_Vector (STORAGE lhs_, const VEC_TYPE &rhs_)`
- `STORAGE operator[] (R_xlen_t i) const`
- `R_xlen_t size () const`

## Private Attributes

- `STORAGE lhs`
- `const VEC_EXT & rhs`
- `bool lhs_na`

### 6.185.1 Detailed Description

```

template<int RTYPE, typename T>
class Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >
  
```

Definition at line 366 of file `divides.h`.

## 6.185.2 Member Typedef Documentation

### 6.185.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T
>::STORAGE
```

Definition at line 370 of file divides.h.

### 6.185.2.2 VEC\_EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor<RTYPE, false, T>::type Rcpp::sugar::Divides_Primitive_Vector< RTYPE,
false, T >::VEC_EXT
```

Definition at line 371 of file divides.h.

### 6.185.2.3 VEC\_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE, false, T> Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T
>::VEC_TYPE
```

Definition at line 369 of file divides.h.

## 6.185.3 Constructor & Destructor Documentation

### 6.185.3.1 Divides\_Primitive\_Vector()

```
template<int RTYPE, typename T >
Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::Divides_Primitive_Vector (
    STORAGE lhs_,
    const VEC_TYPE & rhs_ ) [inline]
```

Definition at line 373 of file divides.h.

## 6.185.4 Member Function Documentation

### 6.185.4.1 operator[]()

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::operator[] (   
    R_xlen_t i ) const [inline]
```

Definition at line 376 of file divides.h.

References `Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::lhs`, `Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::lhs_na`, and `Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::rhs`.

### 6.185.4.2 size()

```
template<int RTYPE, typename T >  
R_xlen_t Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::size ( ) const [inline]
```

Definition at line 380 of file divides.h.

References `Rcpp::sugar::Divides_Primitive_Vector< RTYPE, NA, T >::rhs`.

## 6.185.5 Member Data Documentation

### 6.185.5.1 lhs

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::lhs [private]
```

Definition at line 383 of file divides.h.

### 6.185.5.2 lhs\_na

```
template<int RTYPE, typename T >  
bool Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::lhs_na [private]
```

Definition at line 385 of file divides.h.



### 6.185.5.3 rhs

```
template<int RTYPE, typename T >  
const VEC_EXT& Rcpp::sugar::Divides_Primitive_Vector< RTYPE, false, T >::rhs [private]
```

Definition at line 384 of file divides.h.

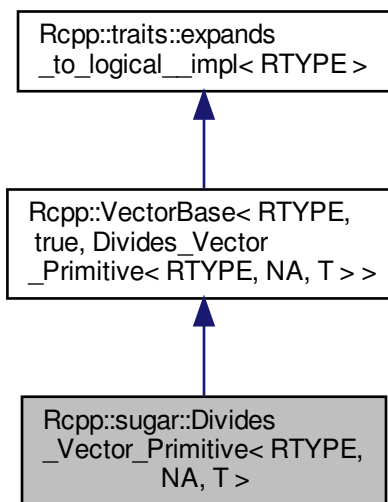
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/divides.h](#)

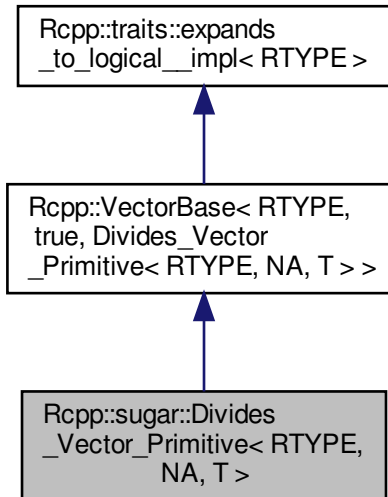
## 6.186 Rcpp::sugar::Divides\_Vector\_Primitive< RTYPE, NA, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Vector\_Primitive< RTYPE, NA, T >:



Collaboration diagram for `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >`:



## Public Types

- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< RTYPE, NA, T >::type` `VEC_EXT`

## Public Member Functions

- `Divides_Vector_Primitive` (const `VEC_TYPE` &lhs, `STORAGE` rhs\_)
- `STORAGE operator[]` (`R_xlen_t` i) const
- `R_xlen_t size` () const

## Private Attributes

- const `VEC_EXT` & lhs
- `STORAGE` rhs
- bool `rhs_na`

### 6.186.1 Detailed Description

```

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >
  
```

Definition at line 226 of file `divides.h`.

## 6.186.2 Member Typedef Documentation

### 6.186.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T
>::STORAGE
```

Definition at line 229 of file divides.h.

### 6.186.2.2 VEC\_EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor<RTYPE,NA,T>::type Rcpp::sugar::Divides_Vector_Primitive< RTYPE,
NA, T >::VEC_EXT
```

Definition at line 231 of file divides.h.

### 6.186.2.3 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >←
::VEC_TYPE
```

Definition at line 230 of file divides.h.

## 6.186.3 Constructor & Destructor Documentation

### 6.186.3.1 Divides\_Vector\_Primitive()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::Divides_Vector_Primitive (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 233 of file divides.h.

## 6.186.4 Member Function Documentation

### 6.186.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 237 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs`, `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::rhs`, and `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::rhs_na`.

### 6.186.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 243 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs`.

## 6.186.5 Member Data Documentation

### 6.186.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs [private]
```

Definition at line 246 of file divides.h.

Referenced by `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::operator[]()`, `Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::operator[]()`, `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::operator[]()`, `Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::operator[]()`, `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::size()`, `Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::size()`, `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::size()`, and `Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::size()`.

### 6.186.5.2 rhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::rhs [private]
```

Definition at line 247 of file divides.h.

Referenced by Rcpp::sugar::Divides\_Vector\_Primitive< RTYPE, NA, T >::operator[](), Rcpp::sugar::Divides\_Vector\_↵\_Primitive< REALSXP, NA, T >::operator[](), Rcpp::sugar::Divides\_Vector\_Primitive< RTYPE, false, T >::operator[](), and Rcpp::sugar::Divides\_Vector\_Primitive< REALSXP, false, T >::operator[]().

### 6.186.5.3 rhs\_na

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::rhs_na [private]
```

Definition at line 248 of file divides.h.

Referenced by Rcpp::sugar::Divides\_Vector\_Primitive< RTYPE, NA, T >::operator[](), and Rcpp::sugar::Divides\_↵\_Vector\_Primitive< RTYPE, false, T >::operator[]().

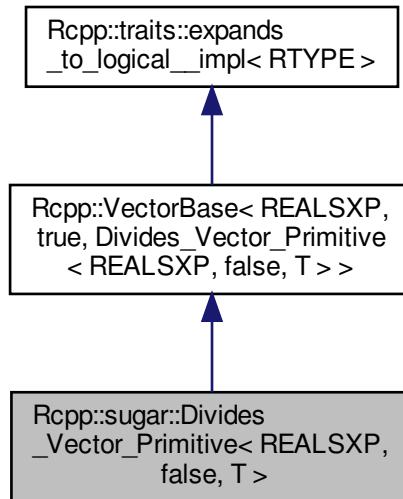
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/divides.h](#)

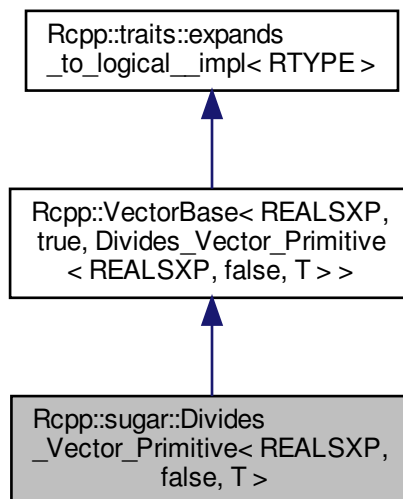
## 6.187 Rcpp::sugar::Divides\_Vector\_Primitive< REALSXP, false, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for `Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >`:



Collaboration diagram for `Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >`:



## Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, false, T > [VEC\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, T >::type [VEC\\_EXT](#)

## Public Member Functions

- [Divides\\_Vector\\_Primitive](#) (const [VEC\\_TYPE](#) &lhs\_, double rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [VEC\\_EXT](#) & lhs
- double rhs

### 6.187.1 Detailed Description

```
template<typename T>
class Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >
```

Definition at line 300 of file divides.h.

### 6.187.2 Member Typedef Documentation

#### 6.187.2.1 VEC\_EXT

```
template<typename T >
typedef Rcpp::traits::Extractor<REALSXP,false,T>::type Rcpp::sugar::Divides\_Vector\_Primitive<
REALSXP, false, T >::VEC_EXT
```

Definition at line 304 of file divides.h.

#### 6.187.2.2 VEC\_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP,false,T> Rcpp::sugar::Divides\_Vector\_Primitive< REALSXP, false,
T >::VEC_TYPE
```

Definition at line 303 of file divides.h.

## 6.187.3 Constructor & Destructor Documentation

### 6.187.3.1 Divides\_Vector\_Primitive()

```
template<typename T >
Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::Divides_Vector_Primitive (
    const VEC_TYPE & lhs_,
    double rhs_ ) [inline]
```

Definition at line 306 of file divides.h.

## 6.187.4 Member Function Documentation

### 6.187.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 309 of file divides.h.

References [Rcpp::sugar::Divides\\_Vector\\_Primitive< RTYPE, NA, T >::lhs](#), and [Rcpp::sugar::Divides\\_Vector\\_Primitive< RTYPE, NA, T >::rhs](#).

### 6.187.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::size ( ) const [inline]
```

Definition at line 312 of file divides.h.

References [Rcpp::sugar::Divides\\_Vector\\_Primitive< RTYPE, NA, T >::lhs](#).

## 6.187.5 Member Data Documentation



## 6.187.5.1 lhs

```
template<typename T >
const VEC_EXT& Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::lhs [private]
```

Definition at line 315 of file divides.h.

## 6.187.5.2 rhs

```
template<typename T >
double Rcpp::sugar::Divides_Vector_Primitive< REALSXP, false, T >::rhs [private]
```

Definition at line 316 of file divides.h.

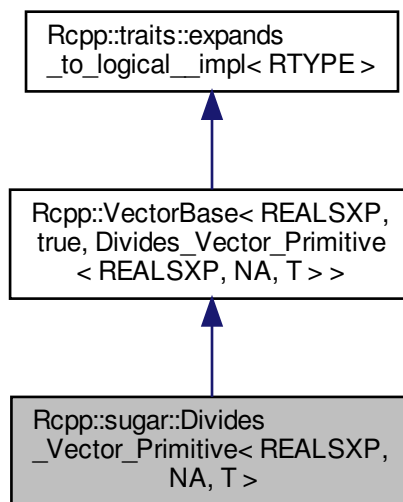
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

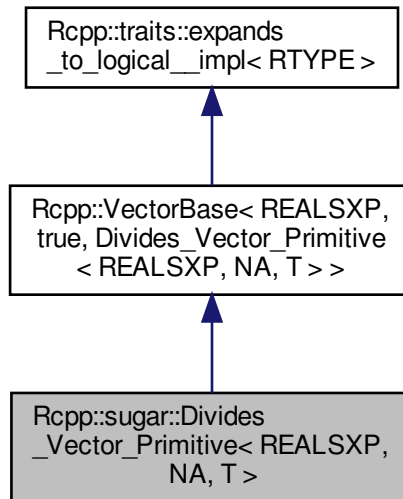
## 6.188 Rcpp::sugar::Divides\_Vector\_Primitive< REALSXP, NA, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Vector\_Primitive< REALSXP, NA, T >:



Collaboration diagram for `Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >`:



## Public Types

- typedef `Rcpp::VectorBase< REALSXP, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< REALSXP, NA, T >::type` `VEC_EXT`

## Public Member Functions

- `Divides_Vector_Primitive` (const `VEC_TYPE` &lhs\_, double rhs\_)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const `VEC_EXT` & lhs
- double rhs

### 6.188.1 Detailed Description

```

template<bool NA, typename T>
class Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >
  
```

Definition at line 252 of file `divides.h`.

## 6.188.2 Member Typedef Documentation

### 6.188.2.1 VEC\_EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP,NA,T>::type Rcpp::sugar::Divides_Vector_Primitive< REALSXP,
NA, T >::VEC_EXT
```

Definition at line 256 of file divides.h.

### 6.188.2.2 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >←
::VEC_TYPE
```

Definition at line 255 of file divides.h.

## 6.188.3 Constructor & Destructor Documentation

### 6.188.3.1 Divides\_Vector\_Primitive()

```
template<bool NA, typename T >
Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::Divides_Vector_Primitive (
    const VEC_TYPE & lhs_,
    double rhs_ ) [inline]
```

Definition at line 258 of file divides.h.

## 6.188.4 Member Function Documentation

### 6.188.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::operator[] (
    R_xlen_t i) const [inline]
```

Definition at line 262 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs`, and `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::rhs`.

### 6.188.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::size ( ) const [inline]
```

Definition at line 266 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs`.

## 6.188.5 Member Data Documentation

### 6.188.5.1 lhs

```
template<bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::lhs [private]
```

Definition at line 269 of file divides.h.

### 6.188.5.2 rhs

```
template<bool NA, typename T >
double Rcpp::sugar::Divides_Vector_Primitive< REALSXP, NA, T >::rhs [private]
```

Definition at line 270 of file divides.h.

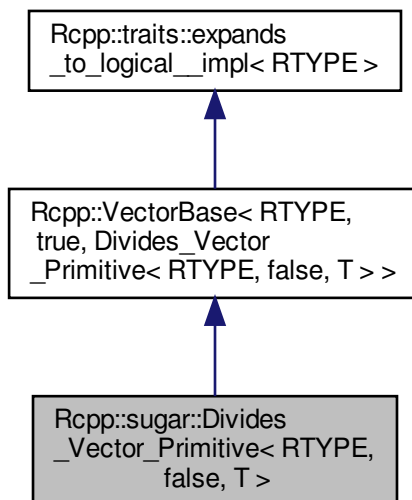
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/operators/divides.h`

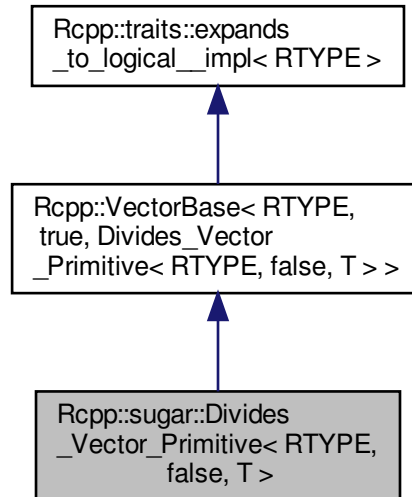
## 6.189 Rcpp::sugar::Divides\_Vector\_Primitive< RTYPE, false, T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Vector\_Primitive< RTYPE, false, T >:



Collaboration diagram for `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >`:



## Public Types

- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::VectorBase< RTYPE, false, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< RTYPE, false, T >::type` `VEC_EXT`

## Public Member Functions

- `Divides_Vector_Primitive` (const `VEC_TYPE` &lhs, `STORAGE` rhs\_)
- `STORAGE operator[]` (`R_xlen_t` i) const
- `R_xlen_t size` () const

## Private Attributes

- const `VEC_EXT` & lhs
- `STORAGE` rhs
- bool rhs\_na

### 6.189.1 Detailed Description

```

template<int RTYPE, typename T>
class Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >
  
```

Definition at line 276 of file `divides.h`.

## 6.189.2 Member Typedef Documentation

### 6.189.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T
>::STORAGE
```

Definition at line 279 of file divides.h.

### 6.189.2.2 VEC\_EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor<RTYPE, false, T>::type Rcpp::sugar::Divides_Vector_Primitive< RTYPE,
false, T >::VEC_EXT
```

Definition at line 281 of file divides.h.

### 6.189.2.3 VEC\_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE, false, T> Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T
>::VEC_TYPE
```

Definition at line 280 of file divides.h.

## 6.189.3 Constructor & Destructor Documentation

### 6.189.3.1 Divides\_Vector\_Primitive()

```
template<int RTYPE, typename T >
Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::Divides_Vector_Primitive (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 283 of file divides.h.

## 6.189.4 Member Function Documentation

### 6.189.4.1 operator[]()

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::operator[] (   
    R_xlen_t i ) const [inline]
```

Definition at line 286 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs`, `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::rhs`, and `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::rhs_na`.

### 6.189.4.2 size()

```
template<int RTYPE, typename T >  
R_xlen_t Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::size ( ) const [inline]
```

Definition at line 291 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Primitive< RTYPE, NA, T >::lhs`.

## 6.189.5 Member Data Documentation

### 6.189.5.1 lhs

```
template<int RTYPE, typename T >  
const VEC_EXT& Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::lhs [private]
```

Definition at line 294 of file divides.h.

### 6.189.5.2 rhs

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::rhs [private]
```

Definition at line 295 of file divides.h.



6.189.5.3 rhs\_na

```
template<int RTYPE, typename T >
bool Rcpp::sugar::Divides_Vector_Primitive< RTYPE, false, T >::rhs_na [private]
```

Definition at line 296 of file divides.h.

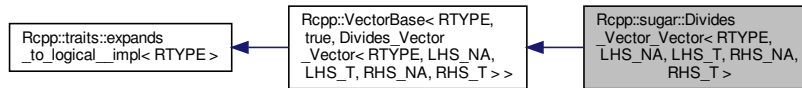
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/divides.h](#)

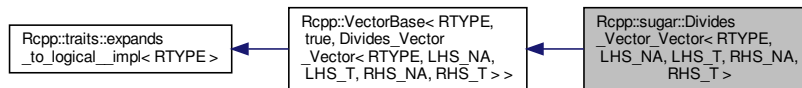
6.190 Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, RHS\_NA, RHS\_T >::type [RHS\\_EXT](#)

## Public Member Functions

- [Divides\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & lhs
- const [RHS\\_EXT](#) & rhs

### 6.190.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 29 of file divides.h.

### 6.190.2 Member Typedef Documentation

#### 6.190.2.1 LHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 34 of file divides.h.

#### 6.190.2.2 LHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE, LHS_NA, LHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 31 of file divides.h.

### 6.190.2.3 RHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 35 of file divides.h.

### 6.190.2.4 RHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 32 of file divides.h.

### 6.190.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 33 of file divides.h.

## 6.190.3 Constructor & Destructor Documentation

### 6.190.3.1 Divides\_Vector\_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Divides_Vector_Vector
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 37 of file divides.h.

### 6.190.4 Member Function Documentation

### 6.190.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 40 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

### 6.190.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( )
const [inline]
```

Definition at line 47 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

## 6.190.5 Member Data Documentation

### 6.190.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 50 of file divides.h.

Referenced by `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[]()`, `Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[]()`, `Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[]()`, `Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[]()`, `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[]()`, `Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[]()`, `Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[]()`, `Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[]()`, `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size()`, `Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size()`, `Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size()`, `Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size()`, `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size()`, `Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size()`, and `Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size()`.

6.190.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 51 of file divides.h.

Referenced by Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[ ](), Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[ ](), Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >::operator[ ](), Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >::operator[ ](), Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >::operator[ ](), Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >::operator[ ](), Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >::operator[ ](), and Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >::operator[ ]().

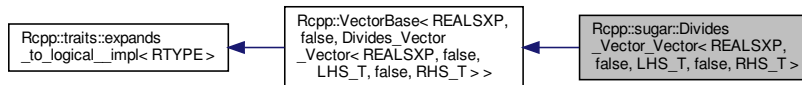
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

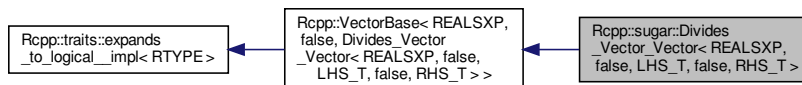
6.191 Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< REALSXP, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, RHS\_T >::type [RHS\\_EXT](#)

## Public Member Functions

- [Divides\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & [lhs](#)
- const [RHS\\_EXT](#) & [rhs](#)

### 6.191.1 Detailed Description

```
template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
```

Definition at line 200 of file divides.h.

### 6.191.2 Member Typedef Documentation

#### 6.191.2.1 LHS\_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Divides\_Vector\_Vector<
REALSXP, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 205 of file divides.h.

### 6.191.2.2 LHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, false, LHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 203 of file divides.h.

### 6.191.2.3 RHS\_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
REALSXP, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 206 of file divides.h.

### 6.191.2.4 RHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, false, RHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 204 of file divides.h.

## 6.191.3 Constructor & Destructor Documentation

### 6.191.3.1 Divides\_Vector\_Vector()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::Divides_Vector_Vector
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 208 of file divides.h.

### 6.191.4 Member Function Documentation

### 6.191.4.1 operator[]()

```
template<typename LHS_T , typename RHS_T >
double Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 211 of file divides.h.

References Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.191.4.2 size()

```
template<typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size ( )
const [inline]
```

Definition at line 215 of file divides.h.

References Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs.

## 6.191.5 Member Data Documentation

### 6.191.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::lhs
[private]
```

Definition at line 218 of file divides.h.

### 6.191.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::rhs
[private]
```

Definition at line 219 of file divides.h.

The documentation for this class was generated from the following file:

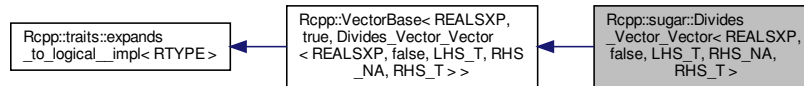
- inst/include/Rcpp/sugar/operators/divides.h



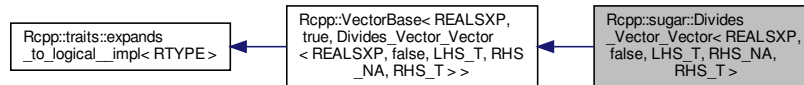
## 6.192 Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase< REALSXP, false, LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase< REALSXP, RHS\\_NA, RHS\\_T >](#) [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor< REALSXP, false, LHS\\_T >::type](#) [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor< REALSXP, RHS\\_NA, RHS\\_T >::type](#) [RHS\\_EXT](#)

### Public Member Functions

- [Divides\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

### 6.192.1 Detailed Description

```
template<typename LHS_T, bool RHS_NA, typename RHS_T>  
class Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
```

Definition at line 104 of file divides.h.

### 6.192.2 Member Typedef Documentation

#### 6.192.2.1 LHS\_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Divides_Vector_Vector<  
REALSXP, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 109 of file divides.h.

#### 6.192.2.2 LHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::VectorBase<REALSXP, false, LHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, false,  
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 107 of file divides.h.

#### 6.192.2.3 RHS\_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<  
REALSXP, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 110 of file divides.h.

### 6.192.2.4 RHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,RHS_NA,RHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 108 of file divides.h.

## 6.192.3 Constructor & Destructor Documentation

### 6.192.3.1 Divides\_Vector\_Vector()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::Divides_Vector_Vector
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 112 of file divides.h.

## 6.192.4 Member Function Documentation

### 6.192.4.1 operator[]()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 115 of file divides.h.

References [Rcpp::sugar::Divides\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#), and [Rcpp::sugar::Divides\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::rhs](#).

### 6.192.4.2 size()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size ( )
const [inline]
```

Definition at line 119 of file divides.h.

References [Rcpp::sugar::Divides\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#).

## 6.192.5 Member Data Documentation

### 6.192.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 122 of file divides.h.

### 6.192.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 123 of file divides.h.

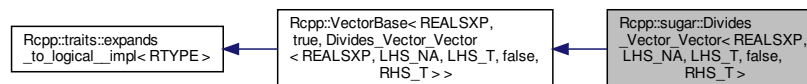
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/divides.h](#)

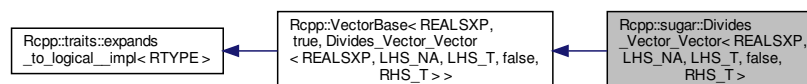
## 6.193 Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< REALSXP, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, RHS\_T >::type [RHS\\_EXT](#)

## Public Member Functions

- [Divides\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & lhs
- const [RHS\\_EXT](#) & rhs

### 6.193.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
```

Definition at line 153 of file divides.h.

### 6.193.2 Member Typedef Documentation

#### 6.193.2.1 LHS\_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Divides\_Vector\_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 158 of file divides.h.

### 6.193.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,LHS_NA,LHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_↵
_NA, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 156 of file divides.h.

### 6.193.2.3 RHS\_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 159 of file divides.h.

### 6.193.2.4 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,RHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_↵
NA, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 157 of file divides.h.

## 6.193.3 Constructor & Destructor Documentation

### 6.193.3.1 Divides\_Vector\_Vector()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::Divides_Vector_Vector
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 161 of file divides.h.

### 6.193.4 Member Function Documentation

#### 6.193.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
double Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 164 of file divides.h.

References Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

#### 6.193.4.2 size()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size ( )
const [inline]
```

Definition at line 167 of file divides.h.

References Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs.

### 6.193.5 Member Data Documentation

#### 6.193.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::lhs
[private]
```

Definition at line 170 of file divides.h.

#### 6.193.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::rhs
[private]
```

Definition at line 171 of file divides.h.

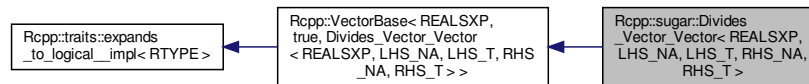
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

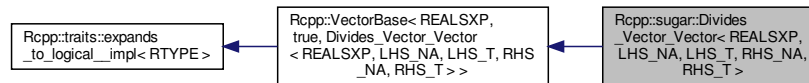
## 6.194 Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Divides\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< REALSXP, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, RHS\_NA, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Divides\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs



## 6.194.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 55 of file divides.h.

## 6.194.2 Member Typedef Documentation

### 6.194.2.1 LHS\_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 60 of file divides.h.

### 6.194.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, LHS_NA, LHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS↔
_NA, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 58 of file divides.h.

### 6.194.2.3 RHS\_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 61 of file divides.h.

### 6.194.2.4 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,RHS_NA,RHS_T> Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS←
_NA, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 59 of file divides.h.

## 6.194.3 Constructor & Destructor Documentation

### 6.194.3.1 Divides\_Vector\_Vector()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::Divides_Vector_Vector
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 63 of file divides.h.

## 6.194.4 Member Function Documentation

### 6.194.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 66 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp←  
::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

### 6.194.4.2 size()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( )
const [inline]
```

Definition at line 70 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

## 6.194.5 Member Data Documentation

### 6.194.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 73 of file divides.h.

### 6.194.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 74 of file divides.h.

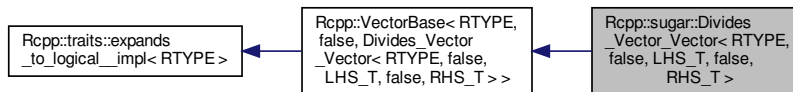
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

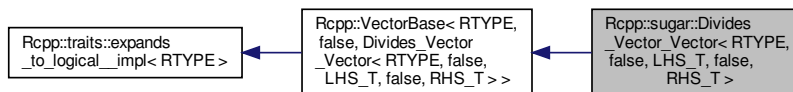
## 6.195 Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, RHS\_T >::type [RHS\\_EXT](#)

## Public Member Functions

- [Divides\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & lhs
- const [RHS\\_EXT](#) & rhs

### 6.195.1 Detailed Description

```
template<int RTYPE, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
```

Definition at line 176 of file divides.h.

### 6.195.2 Member Typedef Documentation

#### 6.195.2.1 LHS\_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, false, LHS_T>::type Rcpp::sugar::Divides\_Vector\_Vector<
RTYPE, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 182 of file divides.h.

### 6.195.2.2 LHS\_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, false,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 179 of file divides.h.

### 6.195.2.3 RHS\_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, false, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 183 of file divides.h.

### 6.195.2.4 RHS\_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, false,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 180 of file divides.h.

### 6.195.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS↔
_T, false, RHS_T >::STORAGE
```

Definition at line 181 of file divides.h.

## 6.195.3 Constructor & Destructor Documentation

### 6.195.3.1 Divides\_Vector\_Vector()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::Divides_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 185 of file divides.h.

## 6.195.4 Member Function Documentation

### 6.195.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 188 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

### 6.195.4.2 size()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 192 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

## 6.195.5 Member Data Documentation

### 6.195.5.1 lhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::lhs [private]
```

Definition at line 195 of file divides.h.

6.195.5.2 rhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::rhs [private]
```

Definition at line 196 of file divides.h.

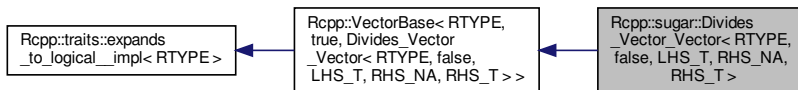
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/divides.h

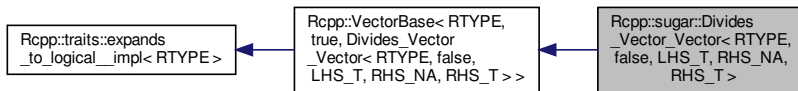
6.196 Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >:



Public Types

- typedef Rcpp::VectorBase< RTYPE, false, LHS\_T > LHS\_TYPE
- typedef Rcpp::VectorBase< RTYPE, RHS\_NA, RHS\_T > RHS\_TYPE
- typedef traits::storage\_type< RTYPE >::type STORAGE
- typedef Rcpp::traits::Extractor< RTYPE, false, LHS\_T >::type LHS\_EXT
- typedef Rcpp::traits::Extractor< RTYPE, RHS\_NA, RHS\_T >::type RHS\_EXT

## Public Member Functions

- [Divides\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) ([R\\_xlen\\_t](#) i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & lhs
- const [RHS\\_EXT](#) & rhs

### 6.196.1 Detailed Description

```
template<int RTYPE, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
```

Definition at line 79 of file divides.h.

### 6.196.2 Member Typedef Documentation

#### 6.196.2.1 LHS\_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 84 of file divides.h.

#### 6.196.2.2 LHS\_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, false,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 81 of file divides.h.



### 6.196.2.3 RHS\_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 85 of file divides.h.

### 6.196.2.4 RHS\_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 82 of file divides.h.

### 6.196.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS↔
_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 83 of file divides.h.

## 6.196.3 Constructor & Destructor Documentation

### 6.196.3.1 Divides\_Vector\_Vector()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::Divides_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 87 of file divides.h.

## 6.196.4 Member Function Documentation

### 6.196.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 90 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

### 6.196.4.2 size()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 96 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

## 6.196.5 Member Data Documentation

### 6.196.5.1 lhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 99 of file divides.h.

### 6.196.5.2 rhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 100 of file divides.h.

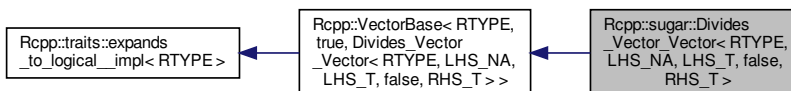
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/operators/divides.h`

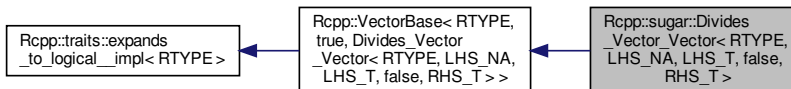
## 6.197 Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <divides.h>
```

Inheritance diagram for Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Divides\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

### 6.197.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
```

Definition at line 128 of file divides.h.

### 6.197.2 Member Typedef Documentation

#### 6.197.2.1 LHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 134 of file divides.h.

#### 6.197.2.2 LHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE, LHS_NA, LHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 131 of file divides.h.

#### 6.197.2.3 RHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T>::type Rcpp::sugar::Divides_Vector_Vector<
RTYPE, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 135 of file divides.h.

### 6.197.2.4 RHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 132 of file divides.h.

### 6.197.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA,
LHS_T, false, RHS_T >::STORAGE
```

Definition at line 133 of file divides.h.

## 6.197.3 Constructor & Destructor Documentation

### 6.197.3.1 Divides\_Vector\_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::Divides_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 137 of file divides.h.

## 6.197.4 Member Function Documentation

### 6.197.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 140 of file divides.h.

References Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Divides\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.197.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 145 of file divides.h.

References `Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

## 6.197.5 Member Data Documentation

### 6.197.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::lhs
[private]
```

Definition at line 148 of file divides.h.

### 6.197.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::rhs
[private]
```

Definition at line 149 of file divides.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/divides.h](#)

## 6.198 Rcpp::DottedPairImpl< CLASS > Class Template Reference

```
#include <DottedPairImpl.h>
```

## Public Member Functions

- `template<typename T >`  
void `push_back` (const T &`object`)
- `template<typename T >`  
void `push_front` (const T &`object`)
- `template<typename T >`  
void `insert` (const `size_t` &`index`, const T &`object`)
- `template<typename T >`  
void `replace` (const `int` &`index`, const T &`object`)
- `R_xlen_t` `length` () const
- `R_xlen_t` `size` () const
- void `remove` (const `size_t` &`index`)

## Friends

- `template<typename T >`  
`DottedPairImpl` & `operator<<` (`DottedPairImpl` &`os`, const T &`t`)
- `template<typename T >`  
`DottedPairImpl` & `operator>>` (const T &`t`, `DottedPairImpl` &`s`)

### 6.198.1 Detailed Description

```
template<typename CLASS>
class Rcpp::DottedPairImpl< CLASS >
```

Definition at line 24 of file `DottedPairImpl.h`.

### 6.198.2 Member Function Documentation

#### 6.198.2.1 insert()

```
template<typename CLASS >
template<typename T >
void Rcpp::DottedPairImpl< CLASS >::insert (
    const size_t & index,
    const T & object )
```

insert an object at the given position, pushing other objects to the tail of the list

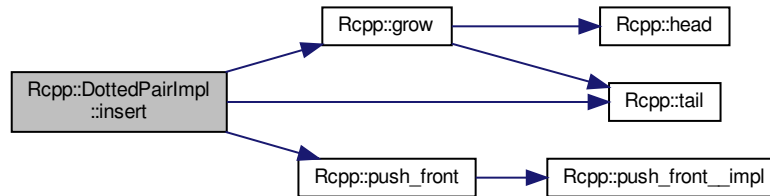
#### Parameters

<i>index</i>	index (0-based) where to insert
<i>object</i>	object to wrap

Definition at line 49 of file DottedPairImpl.h.

References `Rcpp::grow()`, `Rcpp::push_front()`, and `Rcpp::tail()`.

Here is the call graph for this function:



### 6.198.2.2 length()

```
template<typename CLASS >
R_xlen_t Rcpp::DottedPairImpl< CLASS >::length ( ) const [inline]
```

Definition at line 66 of file DottedPairImpl.h.

### 6.198.2.3 push\_back()

```
template<typename CLASS >
template<typename T >
void Rcpp::DottedPairImpl< CLASS >::push_back (
    const T & object )
```

wraps an object and add it at the end of the pairlist (this require traversing the entire pairlist)

#### Parameters

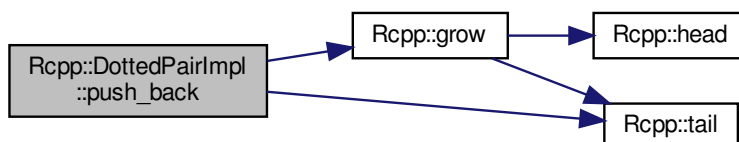
<i>object</i>	anything that can be wrapped by one of the wrap functions, named objects (instances of <code>traits::named_object&lt;&gt;</code> are treated specially)
---------------	---

Definition at line 32 of file DottedPairImpl.h.

References `Rcpp::grow()`, and `Rcpp::tail()`.



Here is the call graph for this function:



#### 6.198.2.4 push\_front()

```

template<typename CLASS >
template<typename T >
void Rcpp::DottedPairImpl< CLASS >::push_front (
    const T & object )
  
```

wraps an object and add it in front of the pairlist.

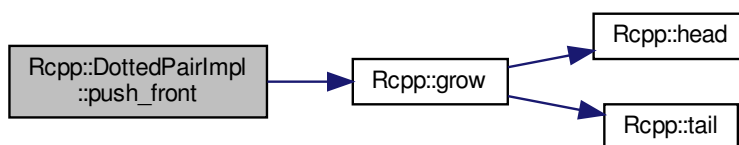
##### Parameters

<i>object</i>	anything that can be wrapped by one of the wrap functions, or an object of class Named
---------------	--

Definition at line 25 of file DottedPairImpl.h.

References `Rcpp::grow()`.

Here is the call graph for this function:



**6.198.2.5 remove()**

```
template<typename CLASS >
void Rcpp::DottedPairImpl< CLASS >::remove (
    const size_t & index )
```

Remove the element at the given position

**Parameters**

<i>index</i>	position where the element is to be removed
--------------	---

Definition at line 101 of file DottedPairImpl.h.

**6.198.2.6 replace()**

```
template<typename CLASS >
template<typename T >
void Rcpp::DottedPairImpl< CLASS >::replace (
    const int & index,
    const T & object )
```

replaces an element of the list

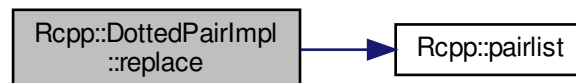
**Parameters**

<i>index</i>	position
<i>object</i>	object that can be wrapped

Definition at line 80 of file DottedPairImpl.h.

References Rcpp::pairlist().

Here is the call graph for this function:



### 6.198.2.7 size()

```
template<typename CLASS >
R_xlen_t Rcpp::DottedPairImpl< CLASS >::size ( ) const [inline]
```

Definition at line 70 of file DottedPairImpl.h.

## 6.198.3 Friends And Related Function Documentation

### 6.198.3.1 operator<<

```
template<typename CLASS >
template<typename T >
DottedPairImpl& operator<< (
    DottedPairImpl< CLASS > & os,
    const T & t ) [friend]
```

Definition at line 82 of file DottedPairImpl.h.

### 6.198.3.2 operator>>

```
template<typename CLASS >
template<typename T >
DottedPairImpl& operator>> (
    const T & t,
    DottedPairImpl< CLASS > & s ) [friend]
```

Definition at line 88 of file DottedPairImpl.h.

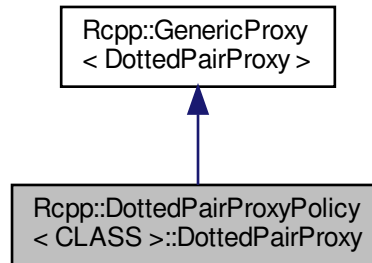
The documentation for this class was generated from the following file:

- inst/include/Rcpp/DottedPairImpl.h

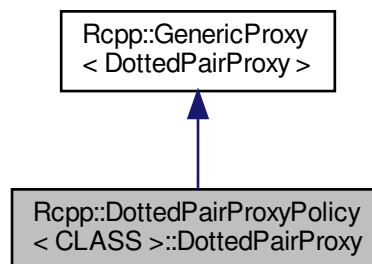
## 6.199 Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy Class Reference

```
#include <DottedPairProxy.h>
```

Inheritance diagram for Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy:



Collaboration diagram for Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy:



### Public Member Functions

- [DottedPairProxy](#) (CLASS &v, int index\_)
- [DottedPairProxy](#) & [operator=](#) (const [DottedPairProxy](#) &rhs)
- [DottedPairProxy](#) & [operator=](#) (SEXP rhs)
- `template<typename T >`  
[DottedPairProxy](#) & [operator=](#) (const T &rhs)

- `template<typename T >`  
`DottedPairProxy & operator=` (const `traits::named_object< T >` &rhs)
- `template<typename T >`  
`operator T` () const
- `SEXP get` () const
- `operator SEXP` () const
- `DottedPairProxy & set` (SEXP x)
- `DottedPairProxy & set` (SEXP x, const char \*name)
- `template<typename T >`  
`DottedPairProxyPolicy< CLASS >::DottedPairProxy & operator=` (const T &rhs)
- `template<typename T >`  
`DottedPairProxyPolicy< CLASS >::DottedPairProxy & operator=` (const `traits::named_object< T >` &rhs)

## Private Attributes

- `SEXP node`

### 6.199.1 Detailed Description

```
template<typename CLASS >
class Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy
```

Definition at line 27 of file `DottedPairProxy.h`.

### 6.199.2 Constructor & Destructor Documentation

#### 6.199.2.1 DottedPairProxy()

```
template<typename CLASS >
Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::DottedPairProxy (
    CLASS & v,
    int index_ ) [inline]
```

Definition at line 29 of file `DottedPairProxy.h`.

References `Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node`.

### 6.199.3 Member Function Documentation

### 6.199.3.1 get()

```
template<typename CLASS >
SEXP Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::get ( ) const [inline]
```

Definition at line 56 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator SEXP(), and Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=().

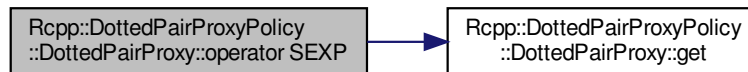
### 6.199.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator SEXP ( ) const [inline]
```

Definition at line 59 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::get().

Here is the call graph for this function:



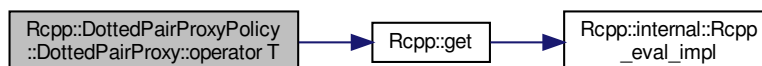
### 6.199.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator T
```

Definition at line 165 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



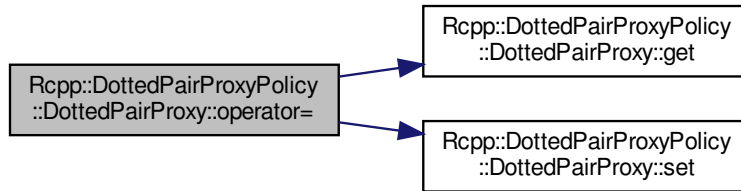
**6.199.3.4 operator=()** [1/6]

```
template<typename CLASS >
DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (
    const DottedPairProxy & rhs ) [inline]
```

Definition at line 41 of file DottedPairProxy.h.

References `Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::get()`, and `Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::set()`.

Here is the call graph for this function:

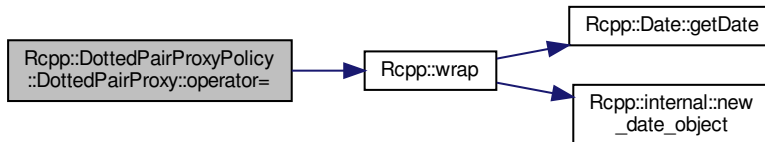
**6.199.3.5 operator=()** [2/6]

```
template<typename CLASS >
template<typename T >
DottedPairProxyPolicy<CLASS>::DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (
    const T & rhs )
```

Definition at line 151 of file proxy.h.

References `Rcpp::wrap()`.

Here is the call graph for this function:



**6.199.3.6 operator=()** [3/6]

```
template<typename CLASS >
template<typename T >
DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (
    const T & rhs )
```

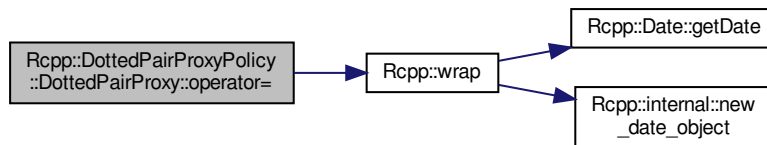
**6.199.3.7 operator=()** [4/6]

```
template<typename CLASS >
template<typename T >
DottedPairProxyPolicy<CLASS>::DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (
    const traits::named_object< T > & rhs )
```

Definition at line 159 of file proxy.h.

References `Rcpp::traits::named_object< T >::name`, `Rcpp::traits::named_object< T >::object`, and `Rcpp::wrap()`.

Here is the call graph for this function:

**6.199.3.8 operator=()** [5/6]

```
template<typename CLASS >
template<typename T >
DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (
    const traits::named_object< T > & rhs )
```



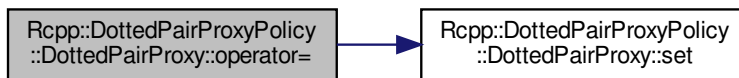
**6.199.3.9 operator=()** [6/6]

```
template<typename CLASS >
DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator= (
    SEXP rhs ) [inline]
```

Definition at line 44 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::set().

Here is the call graph for this function:

**6.199.3.10 set()** [1/2]

```
template<typename CLASS >
DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::set (
    SEXP x ) [inline]
```

Definition at line 62 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=().

**6.199.3.11 set()** [2/2]

```
template<typename CLASS >
DottedPairProxy& Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::set (
    SEXP x,
    const char * name ) [inline]
```

Definition at line 66 of file DottedPairProxy.h.

References Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node.

## 6.199.4 Member Data Documentation

### 6.199.4.1 node

```
template<typename CLASS >
SEXP Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::node [private]
```

Definition at line 74 of file DottedPairProxy.h.

Referenced by `Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::DottedPairProxy()`, `Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::get()`, and `Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::set()`.

The documentation for this class was generated from the following files:

- `inst/include/Rcpp/proxy/DottedPairProxy.h`
- `inst/include/Rcpp/api/meat/proxy.h`

## 6.200 Rcpp::DottedPairProxyPolicy< CLASS > Class Template Reference

```
#include <DottedPairProxy.h>
```

### Classes

- class `const_DottedPairProxy`
- class `DottedPairProxy`

### Public Member Functions

- `DottedPairProxy operator[]` (int i)
- `const_DottedPairProxy operator[]` (int i) const

### 6.200.1 Detailed Description

```
template<typename CLASS>
class Rcpp::DottedPairProxyPolicy< CLASS >
```

Definition at line 24 of file DottedPairProxy.h.

## 6.200.2 Member Function Documentation

### 6.200.2.1 operator[]() [1/2]

```
template<typename CLASS >  
DottedPairProxy Rcpp::DottedPairProxyPolicy< CLASS >::operator[] (   
    int i ) [inline]
```

Definition at line 105 of file DottedPairProxy.h.

### 6.200.2.2 operator[]() [2/2]

```
template<typename CLASS >  
const_DottedPairProxy Rcpp::DottedPairProxyPolicy< CLASS >::operator[] (   
    int i ) const [inline]
```

Definition at line 108 of file DottedPairProxy.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/DottedPairProxy.h

## 6.201 Rcpp::internal::element\_converter< RTYPE > Class Template Reference

```
#include <converter.h>
```

### Public Types

- typedef ::Rcpp::traits::storage\_type< RTYPE >::type [target](#)

### Static Public Member Functions

- template<typename T >  
static [target get](#) (const T &input)
- static [target get](#) (const [target](#) &input)

### 6.201.1 Detailed Description

```
template<int RTYPE>
class Rcpp::internal::element_converter< RTYPE >
```

Definition at line 28 of file converter.h.

### 6.201.2 Member Typedef Documentation

#### 6.201.2.1 target

```
template<int RTYPE>
typedef ::Rcpp::traits::storage_type<RTYPE>::type Rcpp::internal::element_converter< RTYPE >↵
::target
```

Definition at line 30 of file converter.h.

### 6.201.3 Member Function Documentation

#### 6.201.3.1 get() [1/2]

```
template<int RTYPE>
template<typename T >
static target Rcpp::internal::element_converter< RTYPE >::get (
    const T & input ) [inline], [static]
```

Definition at line 33 of file converter.h.

Referenced by `Rcpp::Vector< RTYPE, StoragePolicy >::fill__dispatch()`, `Rcpp::Matrix< RTYPE, StoragePolicy >::fill__diag__dispatch()`, `Rcpp::Vector< RTYPE, StoragePolicy >::insert()`, `Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::operator()`, `Rcpp::sugar::Apply< RTYPE, NA, T, Function, NO_CONVERSION >::operator[]()`, `Rcpp::Vector< RTYPE, StoragePolicy >::push_back()`, `Rcpp::Vector< RTYPE, StoragePolicy >::push__front()`, `Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch()`, and `Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch__isArgument()`.

### 6.201.3.2 get() [2/2]

```
template<int RTYPE>
static target Rcpp::internal::element_converter< RTYPE >::get (
    const target & input ) [inline], [static]
```

Definition at line 37 of file converter.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/[converter.h](#)

## 6.202 Rcpp::traits::enable\_if< B, T > Struct Template Reference

```
#include <enable_if.h>
```

### 6.202.1 Detailed Description

```
template<bool B, typename T = void>
struct Rcpp::traits::enable_if< B, T >
```

Definition at line 8 of file enable\_if.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[enable\\_if.h](#)

## 6.203 Rcpp::traits::enable\_if< true, T > Struct Template Reference

```
#include <enable_if.h>
```

### Public Types

- typedef T [type](#)

### 6.203.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::enable_if< true, T >
```

Definition at line 11 of file enable\_if.h.

## 6.203.2 Member Typedef Documentation

### 6.203.2.1 type

```
template<typename T >  
typedef T Rcpp::traits::enable_if< true, T >::type
```

Definition at line 12 of file `enable_if.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/enable_if.h`

## 6.204 Rcpp::enum\_ < Enum, Parent > Class Template Reference

```
#include <Module.h>
```

### Public Types

- typedef `enum_ < Enum, Parent >` `self`

### Public Member Functions

- `enum_ (const char *name_)`
- `~enum_ ()`
- `self & value (const char *name_, Enum value_)`

### Private Types

- typedef `std::map< std::string, int >` `MAP`
- typedef `MAP::value_type` `PAIR`

### Private Attributes

- `std::string name`
- `MAP values`
- `std::string parent_typeinfo_name`

## 6.204.1 Detailed Description

```
template<typename Enum, typename Parent>  
class Rcpp::enum_< Enum, Parent >
```

Definition at line 343 of file Module.h.

## 6.204.2 Member Typedef Documentation

### 6.204.2.1 MAP

```
template<typename Enum , typename Parent >  
typedef std::map< std::string, int > Rcpp::enum_< Enum, Parent >::MAP [private]
```

Definition at line 363 of file Module.h.

### 6.204.2.2 PAIR

```
template<typename Enum , typename Parent >  
typedef MAP::value_type Rcpp::enum_< Enum, Parent >::PAIR [private]
```

Definition at line 364 of file Module.h.

### 6.204.2.3 self

```
template<typename Enum , typename Parent >  
typedef enum_<Enum,Parent> Rcpp::enum_< Enum, Parent >::self
```

Definition at line 345 of file Module.h.

## 6.204.3 Constructor & Destructor Documentation

### 6.204.3.1 enum\_()

```
template<typename Enum , typename Parent >
Rcpp::enum_< Enum, Parent >::enum_ (
    const char * name_ ) [inline]
```

Definition at line 347 of file Module.h.

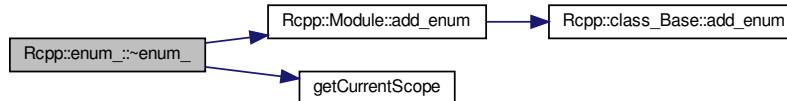
### 6.204.3.2 ~enum\_()

```
template<typename Enum , typename Parent >
Rcpp::enum_< Enum, Parent >::~enum_ ( ) [inline]
```

Definition at line 350 of file Module.h.

References `Rcpp::Module::add_enum()`, `getCurrentScope()`, `Rcpp::enum_< Enum, Parent >::name`, `Rcpp::enum_< Enum, Parent >::parent_typeinfo_name`, and `Rcpp::enum_< Enum, Parent >::values`.

Here is the call graph for this function:



## 6.204.4 Member Function Documentation

### 6.204.4.1 value()

```
template<typename Enum , typename Parent >
self& Rcpp::enum_< Enum, Parent >::value (
    const char * name_,
    Enum value_ ) [inline]
```

Definition at line 355 of file Module.h.

References `Rcpp::enum_< Enum, Parent >::values`.



## 6.204.5 Member Data Documentation

### 6.204.5.1 name

```
template<typename Enum , typename Parent >  
std::string Rcpp::enum_< Enum, Parent >::name [private]
```

Definition at line 362 of file Module.h.

Referenced by Rcpp::enum\_< Enum, Parent >::~~enum\_().

### 6.204.5.2 parent\_typeinfo\_name

```
template<typename Enum , typename Parent >  
std::string Rcpp::enum_< Enum, Parent >::parent_typeinfo_name [private]
```

Definition at line 366 of file Module.h.

Referenced by Rcpp::enum\_< Enum, Parent >::~~enum\_().

### 6.204.5.3 values

```
template<typename Enum , typename Parent >  
MAP Rcpp::enum_< Enum, Parent >::values [private]
```

Definition at line 365 of file Module.h.

Referenced by Rcpp::enum\_< Enum, Parent >::value(), and Rcpp::enum\_< Enum, Parent >::~~enum\_().

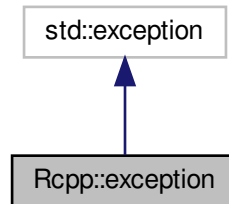
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Module.h](#)

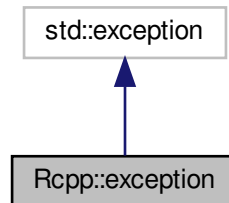
## 6.205 Rcpp::exception Class Reference

```
#include <exceptions.h>
```

Inheritance diagram for Rcpp::exception:



Collaboration diagram for Rcpp::exception:



### Public Member Functions

- [exception](#) (const char \*message\_, bool [include\\_call](#)=RCPP\_DEFAULT\_INCLUDE\_CALL)
- [exception](#) (const char \*message\_, const char \*, int, bool [include\\_call](#)=RCPP\_DEFAULT\_INCLUDE\_CALL)
- bool [include\\_call](#) () const
- virtual [~exception](#) () throw ()
- virtual const char \* [what](#) () const throw ()
- void [copy\\_stack\\_trace\\_to\\_r](#) () const

### Private Member Functions

- void [record\\_stack\\_trace](#) ()

## Private Attributes

- `std::string` `message`
- `bool` `include_call_`
- `std::vector< std::string >` `stack`

### 6.205.1 Detailed Description

Definition at line 36 of file exceptions.h.

### 6.205.2 Constructor & Destructor Documentation

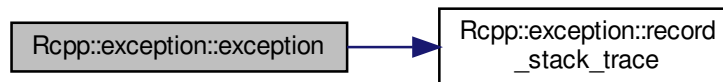
#### 6.205.2.1 exception() [1/2]

```
Rcpp::exception::exception (  
    const char * message_,  
    bool include_call = RCPP_DEFAULT_INCLUDE_CALL ) [inline], [explicit]
```

Definition at line 38 of file exceptions.h.

References `record_stack_trace()`.

Here is the call graph for this function:



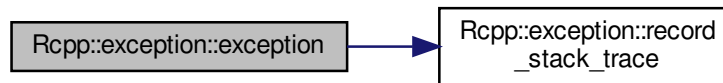
### 6.205.2.2 exception() [2/2]

```
Rcpp::exception::exception (
    const char * message_,
    const char * ,
    int ,
    bool include_call = RCPP_DEFAULT_INCLUDE_CALL ) [inline]
```

Definition at line 43 of file exceptions.h.

References record\_stack\_trace().

Here is the call graph for this function:



### 6.205.2.3 ~exception()

```
virtual Rcpp::exception::~~exception ( ) throw ( ) [inline], [virtual]
```

Definition at line 51 of file exceptions.h.

## 6.205.3 Member Function Documentation

### 6.205.3.1 copy\_stack\_trace\_to\_r()

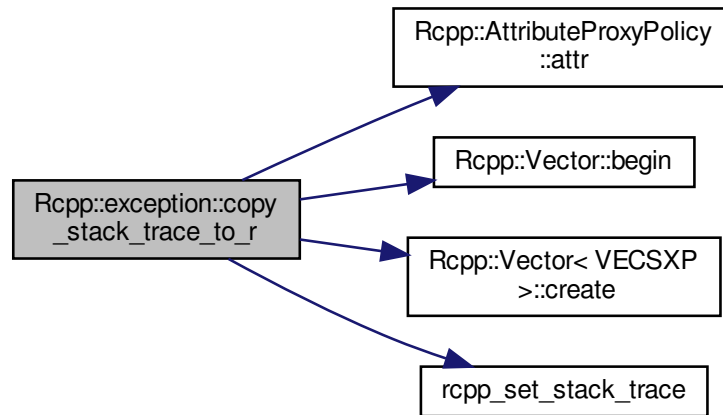
```
void Rcpp::exception::copy_stack_trace_to_r ( ) const [inline]
```

Definition at line 92 of file exceptions\_impl.h.

References Rcpp::\_, Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< VECSXP >::create(), rcpp\_set\_stack\_trace(), and stack.

Referenced by rcpp\_exception\_to\_r\_condition().

Here is the call graph for this function:



### 6.205.3.2 include\_call()

```
bool Rcpp::exception::include_call ( ) const [inline]
```

Definition at line 48 of file exceptions.h.

References include\_call\_.

Referenced by rcpp\_exception\_to\_r\_condition().

### 6.205.3.3 record\_stack\_trace()

```
void Rcpp::exception::record_stack_trace ( ) [inline], [private]
```

Definition at line 74 of file exceptions\_impl.h.

References stack.

Referenced by exception().

### 6.205.3.4 what()

```
virtual const char* Rcpp::exception::what ( ) const throw ( ) [inline], [virtual]
```

Definition at line 52 of file exceptions.h.

References message.

## 6.205.4 Member Data Documentation

### 6.205.4.1 include\_call\_

```
bool Rcpp::exception::include_call_ [private]
```

Definition at line 58 of file exceptions.h.

Referenced by include\_call().

### 6.205.4.2 message

```
std::string Rcpp::exception::message [private]
```

Definition at line 57 of file exceptions.h.

Referenced by what().

### 6.205.4.3 stack

```
std::vector<std::string> Rcpp::exception::stack [private]
```

Definition at line 59 of file exceptions.h.

Referenced by `copy_stack_trace_to_r()`, and `record_stack_trace()`.

The documentation for this class was generated from the following files:

- [inst/include/Rcpp/exceptions.h](#)
- [inst/include/Rcpp/exceptions\\_impl.h](#)

## 6.206 Rcpp::algorithm::helpers::exp Struct Reference

```
#include <algorithm.h>
```

### Public Member Functions

- `template<typename T >`  
double `operator()` (T val)

### 6.206.1 Detailed Description

Definition at line 206 of file algorithm.h.

### 6.206.2 Member Function Documentation

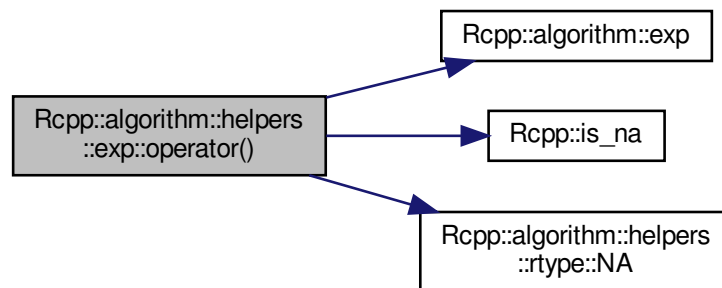
### 6.206.2.1 operator()

```
template<typename T >
double Rcpp::algorithm::helpers::exp::operator() (
    T val ) [inline]
```

Definition at line 208 of file algorithm.h.

References `Rcpp::algorithm::exp()`, `Rcpp::is_na()`, and `Rcpp::algorithm::helpers::rtype< T >::NA()`.

Here is the call graph for this function:



The documentation for this struct was generated from the following file:

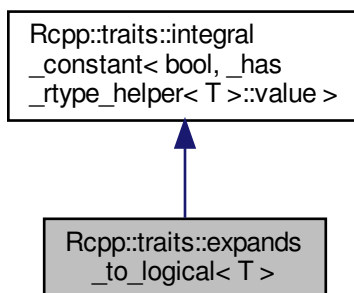
- [inst/include/Rcpp/algorithm.h](#)

## 6.207 Rcpp::traits::expands\_to\_logical< T > Struct Template Reference

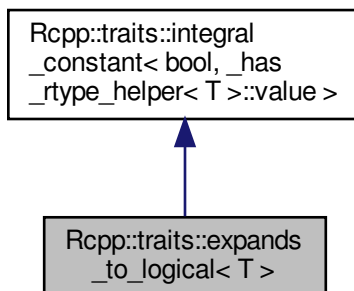
```
#include <expands_to_logical.h>
```



Inheritance diagram for Rcpp::traits::expands\_to\_logical< T >:



Collaboration diagram for Rcpp::traits::expands\_to\_logical< T >:



## Additional Inherited Members

### 6.207.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::expands_to_logical< T >
```

Definition at line 54 of file expands\_to\_logical.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/expands\_to\_logical.h

## 6.208 Rcpp::traits::expands\_to\_logical\_impl< RTYPE > Struct Template Reference

```
#include <expands_to_logical.h>
```

Inherited by [Rcpp::MatrixBase< LGLSXP, false, UpperTri< RTYPE, NA, T > >](#), [Rcpp::MatrixBase< RTYPE, true, SubMatrix< RTYPE >](#), [Rcpp::MatrixBase< INTSXP, false, Row< RTYPE, LHS\\_NA, LHS\\_T > >](#), [Rcpp::MatrixBase< RTYPE, NA, Diag\\_Maker< RTYPE, NA, T >](#), [Rcpp::MatrixBase< INTSXP, false, Col< RTYPE, LHS\\_NA, LHS\\_T > >](#), [Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, ScalarBindable< scalar< RTYPE >::type >, E2 > >](#), [Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, ScalarBindable< scalar< RTYPE >::type >, E2 > >](#), [Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, E1, E2 > >](#), [Rcpp::MatrixBase< RTYPE, true, Matrix< RTYPE, PreserveStorage >](#), [Rcpp::MatrixBase< LGLSXP, false, LowerTri< RTYPE, NA, T > >](#), [Rcpp::MatrixBase< Rcpp::traits::r\\_sexptype\\_traits< ::Rcpp::traits::result\\_of< Function >::type >, true, Mapply\\_2< RTYPE, NA, T > >](#), [Rcpp::MatrixBase< RTYPE, true, JoinOp< RTYPE, E1, E2 > >](#), [Rcpp::MatrixBase< RTYPE, true, Matrix< RTYPE, PreserveStorage >](#), [Rcpp::MatrixBase< RTYPE, NA, T >](#), [Rcpp::VectorBase< RTYPE, true, Times\\_Vector\\_Primitive< RTYPE, NA, T > >](#), [Rcpp::VectorBase< Rcpp::traits::r\\_sexptype\\_traits< RESULT\\_TYPE >::rtype, NA, SugarBlock\\_3\\_VVV< NA, RESULT\\_TYPE, U1, T1 > >](#), [Rcpp::VectorBase< RTYPE, true, Minus\\_Vector\\_Vector< RTYPE, false, LHS\\_T, RHS\\_NA, RHS\\_T > >](#), [Rcpp::VectorBase< RTYPE, true, Minus\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, false, RHS\\_T > >](#), [Rcpp::VectorBase< RTYPE,\(COND\\_NA||LHS\\_NA||RHS\\_NA\), IfElse< RTYPE, COND\\_NA, COND\\_T, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T > >](#), [Rcpp::VectorBase< LGLSXP, true, Comparator< RTYPE, Operator, false, LHS\\_T, false, RHS\\_T > >](#), [Rcpp::VectorBase< REALSXP, true, Divides\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, false, RHS\\_T > >](#), [Rcpp::VectorBase< RTYPE, true, Vector< RTYPE, PreserveStorage >](#), [Rcpp::VectorBase< REALSXP, false, Times\\_Vector\\_Primitive< RTYPE, NA, T > >](#), [Rcpp::VectorBase< RTYPE, NA, Rep< RTYPE, NA, T > >](#), [Rcpp::VectorBase< RTYPE, true, Pmin\\_Vector\\_Primitive< RTYPE, LHS\\_NA, RHS\\_NA >](#), [Rcpp::VectorBase< RTYPE, false, Plus\\_Vector\\_Vector< RTYPE, false, LHS\\_T, false, RHS\\_T > >](#), [Rcpp::VectorBase< RTYPE, false, Plus\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, false, RHS\\_T > >](#), [Rcpp::VectorBase< REALSXP, true, Divides\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, false, RHS\\_T > >](#), [Rcpp::VectorBase< RTYPE, true, IfElse\\_Vector\\_Primitive< RTYPE, COND\\_NA, COND\\_T, LHS\\_NA, LHS\\_T > >](#), [Rcpp::VectorBase< Rcpp::traits::r\\_sexptype\\_traits< RESULT\\_TYPE >::rtype, NA, SugarBlock\\_1< NA, RESULT\\_TYPE, U1, T1 > >](#), [Rcpp::VectorBase< RTYPE, NA, Rep\\_each< RTYPE, NA, T > >](#), [Rcpp::VectorBase< RTYPE,\(LHS\\_NA||RHS\\_NA\), Pmax\\_Vector\\_Vector< RTYPE, LHS\\_NA, RHS\\_NA >](#), [Rcpp::VectorBase< LGLSXP, true, Or\\_LogicalExpression\\_LogicalExpression< false, LHS\\_T, RHS\\_NA, RHS\\_T > >](#), [Rcpp::VectorBase< REALSXP, true, Minus\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T > >](#), [Rcpp::VectorBase< REALSXP, true, Divides\\_Primitive\\_Vector< REALSXP, false, T > >](#), [Rcpp::VectorBase< REALSXP, true, Divides\\_Primitive\\_Vector< REALSXP, NA, P3< RTYPE, NA, T > >](#), [Rcpp::VectorBase< Rcpp::traits::r\\_sexptype\\_traits< T >::rtype, true, Rep\\_each< RTYPE, NA, T > >](#), [Rcpp::VectorBase< REALSXP, NA, Pow< RTYPE, NA, T, EXPONENT\\_TYPE > >](#), [Rcpp::VectorBase< RTYPE, true, Pmax\\_Vector\\_Primitive< RTYPE, LHS\\_NA, RHS\\_NA >](#), [Rcpp::VectorBase< RTYPE, true, IfElse\\_Vector\\_Primitive< RTYPE, COND\\_NA, COND\\_T, LHS\\_NA, LHS\\_T > >](#), [Rcpp::VectorBase< RTYPE, true, IfElse\\_Primitive\\_Vector< RTYPE, false, COND\\_T, RHS\\_NA, RHS\\_T > >](#), [Rcpp::VectorBase< RTYPE, true, Divides\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, false, RHS\\_T > >](#), [Rcpp::VectorBase< RTYPE, true, Divides\\_Vector\\_Vector< REALSXP, false, Vectorized\\_INTSXP< Func, false, VEC > >](#), [Rcpp::VectorBase< RTYPE, true, Times\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, false, RHS\\_T > >](#), [Rcpp::VectorBase< RTYPE, true, Minus\\_Vector\\_Primitive< RTYPE, false, T > >](#), [Rcpp::VectorBase< REALSXP, true, Divides\\_Vector\\_Primitive< RTYPE, true, Divides\\_Vector\\_Primitive< RTYPE, NA, T > >](#), [Rcpp::VectorBase< LGLSXP, true, And\\_LogicalExpression\\_LogicalExpression< false, LHS\\_T, RHS\\_NA, RHS\\_T > >](#), [Rcpp::VectorBase< INTSXP, false, Range >](#), [Rcpp::VectorBase< RTYPE, true, ConstMatrixRow< RTYPE > >](#), [Rcpp::VectorBase< RTYPE, false, Times\\_Vector\\_Primitive< RTYPE, false, T > >](#), [Rcpp::VectorBase< RTYPE, NA, Rep\\_len< RTYPE, NA, T > >](#), [Rcpp::VectorBase< REALSXP, false, Pow< INTSXP, false, T, EXPONENT\\_TYPE > >](#), [Rcpp::VectorBase< REALSXP, true, Plus\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, false, RHS\\_T > >](#), [Rcpp::VectorBase< REALSXP, true, Minus\\_Vector\\_Primitive< RTYPE, NA, T > >](#), [Rcpp::VectorBase< REALSXP, true, Minus\\_Primitive\\_Vector< REALSXP, NA, T > >](#), [Rcpp::VectorBase< LGLSXP, false, IsNaN< RTYPE, NA, VEC\\_TYPE > >](#), [Rcpp::VectorBase< RTYPE, true, Divides\\_Vector\\_Vector< REALSXP, NA, Diag\\_Extractor< RTYPE, NA, T > >](#), [Rcpp::VectorBase< RTYPE, NA, RangeIndexer< RTYPE, NA, VEC\\_TYPE >](#), [Rcpp::VectorBase< RTYPE, true, Plus\\_Vector\\_Vector< RTYPE, false, LHS\\_T, RHS\\_NA, RHS\\_T > >](#), [Rcpp::VectorBase< REALSXP, false, Plus\\_Vector\\_Vector< REALSXP, NA, Head< RTYPE, NA, T > >](#), [Rcpp::VectorBase< REALSXP, true, Divides\\_Vector\\_Primitive< REALSXP, NA, T > >](#), [Rcpp::VectorBase< REALSXP, NA, Vectorized\\_INTSXP< Func, NA, VEC > >](#), [Rcpp::VectorBase< Rcpp::traits::r\\_sexptype\\_traits< RESULT\\_TYPE >::rtype, true, Mapply\\_2< RTYPE, NA, T > >](#), [Rcpp::VectorBase< INTSXP, NA, Sign< RTYPE, NA, T > >](#), [Rcpp::VectorBase< RTYPE, NA, Rev< RTYPE, NA, T > >](#), [Rcpp::VectorBase< RTYPE, false, Plus\\_Vector\\_Primitive\\_nona< RTYPE, false, T > >](#), [Rcpp::VectorBase< RTYPE, true, Plus\\_Vector\\_Primitive< RTYPE, true, Divides\\_Vector\\_Primitive< RTYPE, false, T > >](#), [Rcpp::VectorBase< RTYPE, true, Divides\\_Primitive\\_Vector< RTYPE, true, Divides\\_Primitive\\_Vector< RTYPE, NA, T > >](#), [Rcpp::VectorBase< RTYPE, false, Diff< RTYPE, false, T > >](#), [Rcpp::VectorBase< REALSXP, true, Times\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T > >](#), [Rcpp::VectorBase< REALSXP, true, Plus\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, false, RHS\\_T > >](#), [Rcpp::VectorBase< RTYPE, true, Plus\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, false, RHS\\_T > >](#), [Rcpp::VectorBase< LGLSXP, false, Or\\_LogicalExpression\\_LogicalExpression< false, LHS\\_T, false, RHS\\_T > >](#), [Rcpp::VectorBase< RTYPE, false, Nona< RTYPE, NA, Rcpp::Vector< RTYPE > > >](#), [Rcpp::VectorBase< Rcpp::traits::r\\_sexptype\\_traits< ::Rcpp::traits::result\\_of< Function >::type >, true, Mapply\\_2< RTYPE, NA, T > >](#), [Rcpp::VectorBase< VECSXP, true, Lapply< RTYPE, NA, T, Function > >](#), [Rcpp::VectorBase< RTYPE, true, IfElse\\_Primitive\\_Vector< RTYPE, LHS\\_NA, RHS\\_NA >](#), [Rcpp::VectorBase< LGLSXP, true, And\\_LogicalExpression\\_LogicalExpression< LHS\\_NA, LHS\\_T, false, RHS\\_T > >](#),

```

Rcpp::VectorBase< LGLSXP, true, And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > >,
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::sugar::sapply_application_result_of< Function, T >::type >::rtype, true, Sa
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< ::Rcpp::sugar::sapply_application_result_of< Function, T >::type >::rtype, true, Sa
Rcpp::VectorBase< RTYPE,(LHS_NA||RHS_NA), Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > >,
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< LGLSXP, false,
Rcpp::VectorBase< RTYPE, true, ConstMatrixColumn< RTYPE > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TY
Rcpp::VectorBase< REALSXP, false, Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REALSXP,
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Primitive_nona< REALSXP, NA, T > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_
Rcpp::VectorBase< LGLSXP, false, IsNa< RTYPE, false, VEC_TYPE > >, Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Primitive
Rcpp::VectorBase< RTYPE, LHS_NA, Diff< RTYPE, LHS_NA, LHS_T > >, Rcpp::VectorBase< RTYPE, NA, Clamp_Primitive_Vector_
Rcpp::VectorBase< RTYPE, true, MatrixColumn< RTYPE > >, Rcpp::VectorBase< RTYPE, true, Times_Vector_Vector< RTYPE, LHS
Rcpp::VectorBase< RTYPE, false, Times_Vector_Primitive_nona< RTYPE, false, T > >, Rcpp::VectorBase< REALSXP, true, Times_Ve
Rcpp::VectorBase< RTYPE, true, Times_Vector_Primitive_nona< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, Pow< INTS
Rcpp::VectorBase< REALSXP, LHS_NA, Diff< REALSXP, LHS_NA, LHS_T > >, Rcpp::VectorBase< LGLSXP, true, Comparator_With
Rcpp::VectorBase< REALSXP, NA, Q0< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, D0< RTYPE, NA, T > >,
Rcpp::VectorBase< REALSXP, NA, Vectorized< Func, NA, VEC > >, Rcpp::VectorBase< REALSXP, false, Times_Vector_Vector< REA
Rcpp::VectorBase< REALSXP, false, Times_Vector_Primitive_nona< REALSXP, false, T > >, Rcpp::VectorBase< Rcpp::traits::r_sexpty
Rcpp::VectorBase< INTSXP, false, SeqLen >, Rcpp::VectorBase< RTYPE, false, Minus_Vector_Vector< RTYPE, false, LHS_T, false, R
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE
Rcpp::VectorBase< RTYPE,(LHS_NA||RHS_NA), IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T > >,
Rcpp::VectorBase< REALSXP, true, Divides_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< REA
Rcpp::VectorBase< REALSXP, true, Times_Vector_Primitive< REALSXP, NA, T > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits
Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits< RESULT_TYPE >::rtype, NA, SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REAL
Rcpp::VectorBase< RTYPE, true, Minus_Primitive_Vector< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, Q2< RTYPE, NA,
Rcpp::VectorBase< REALSXP, NA, D2< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector< REALSXP, fa
Rcpp::VectorBase< REALSXP, true, Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T > >, Rcpp::VectorBase< Rcpp:
Rcpp::VectorBase< REALSXP, true, Minus_Vector_Primitive< REALSXP, false, T > >, Rcpp::VectorBase< Rcpp::traits::r_sexptype_tra
Rcpp::VectorBase< LGLSXP, false, IsNa< RTYPE, NA, VEC_TYPE > >, Rcpp::VectorBase< LGLSXP, false, IsInfinite< RTYPE, NA, V
Rcpp::VectorBase< LGLSXP, false, IsFinite< RTYPE, NA, VEC_TYPE > >, Rcpp::VectorBase< RTYPE, false, Divides_Vector_Vector<
Rcpp::VectorBase< LGLSXP, true, Comparator< RTYPE, Operator, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< LGLSXP,
Rcpp::VectorBase< REALSXP, NA, Q1< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, P0< RTYPE, NA, T > >,
Rcpp::VectorBase< REALSXP, NA, D1< RTYPE, NA, T > >, Rcpp::VectorBase< RTYPE, NA, Tail< RTYPE, NA, T > >,
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T > >,
Rcpp::VectorBase< RTYPE, true, Plus_Vector_Primitive< RTYPE, NA, T > >, Rcpp::VectorBase< LGLSXP, NA, Not_Vector< RTYPE,
Rcpp::VectorBase< REALSXP, true, Minus_Primitive_Vector< REALSXP, false, T > >, Rcpp::VectorBase< RTYPE, true, Comparator_V
Rcpp::VectorBase< LGLSXP, false, And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T > >,
Rcpp::VectorBase< unary_minus_result_type< RTYPE >::value, NA, UnaryMinus_Vector< unary_minus_result_type< RTYPE >::valu
Rcpp::VectorBase< RTYPE, false, Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< Rcpp::traits::r_
Rcpp::VectorBase< REALSXP, true, Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T > >, Rcpp::VectorBase< RTYPE
Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T > >,
Rcpp::VectorBase< LGLSXP, true, Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > >,
Rcpp::VectorBase< REALSXP, false, Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< RTYPE, t
Rcpp::VectorBase< RTYPE, true, Minus_Primitive_Vector< RTYPE, false, T > >, Rcpp::VectorBase< RTYPE, true, IfElse_Primitive_Pr
Rcpp::VectorBase< REALSXP, false, Divides_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T > >, Rcpp::VectorBase< REALS
Rcpp::VectorBase< REALSXP, NA, P2< RTYPE, NA, T > >, Rcpp::VectorBase< REALSXP, NA, D3< RTYPE, NA, T > >,
Rcpp::MatrixBase< RTYPE, na, MATRIX >, and Rcpp::VectorBase< RTYPE, na, VECTOR >.

```

### 6.208.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::traits::expands_to_logical_impl< RTYPE >
```

Definition at line 33 of file expands\_to\_logical.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/expands\\_to\\_logical.h](#)

## 6.209 Rcpp::traits::expands\_to\_logical\_impl< LGLSXP > Struct Reference

```
#include <expands_to_logical.h>
```

### Classes

- struct [r\\_expands\\_to\\_logical](#)

### 6.209.1 Detailed Description

Definition at line 36 of file expands\_to\_logical.h.

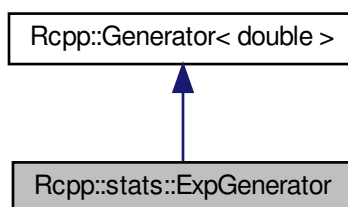
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/expands\\_to\\_logical.h](#)

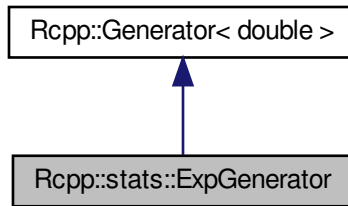
## 6.210 Rcpp::stats::ExpGenerator Class Reference

```
#include <rexp.h>
```

Inheritance diagram for Rcpp::stats::ExpGenerator:



Collaboration diagram for Rcpp::stats::ExpGenerator:



## Public Member Functions

- [ExpGenerator](#) (double scale\_)
- double [operator\(\)](#) () const

## Private Attributes

- double [scale](#)

## Additional Inherited Members

### 6.210.1 Detailed Description

Definition at line 28 of file rexp.h.

### 6.210.2 Constructor & Destructor Documentation

#### 6.210.2.1 ExpGenerator()

```
Rcpp::stats::ExpGenerator::ExpGenerator (  
    double scale_ ) [inline]
```

Definition at line 31 of file rexp.h.

### 6.210.3 Member Function Documentation

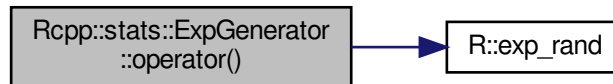
#### 6.210.3.1 operator()

```
double Rcpp::stats::ExpGenerator::operator() ( ) const [inline]
```

Definition at line 33 of file rexp.h.

References `R::exp_rand()`, and `scale`.

Here is the call graph for this function:



### 6.210.4 Member Data Documentation

#### 6.210.4.1 scale

```
double Rcpp::stats::ExpGenerator::scale [private]
```

Definition at line 38 of file rexp.h.

Referenced by `operator()`.

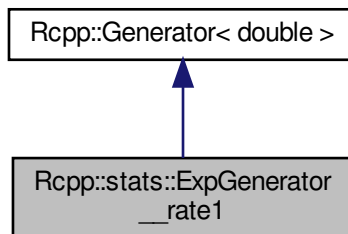
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rexp.h](#)

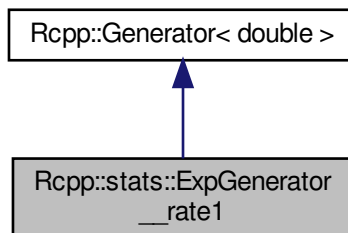
## 6.211 Rcpp::stats::ExpGenerator\_\_rate1 Class Reference

```
#include <rexp.h>
```

Inheritance diagram for Rcpp::stats::ExpGenerator\_\_rate1:



Collaboration diagram for Rcpp::stats::ExpGenerator\_\_rate1:



### Public Member Functions

- `ExpGenerator__rate1 ()`
- `double operator() () const`

### Additional Inherited Members

#### 6.211.1 Detailed Description

Definition at line 41 of file `rexp.h`.

## 6.211.2 Constructor & Destructor Documentation

### 6.211.2.1 ExpGenerator\_\_rate1()

```
Rcpp::stats::ExpGenerator__rate1::ExpGenerator__rate1 ( ) [inline]
```

Definition at line 43 of file rexp.h.

## 6.211.3 Member Function Documentation

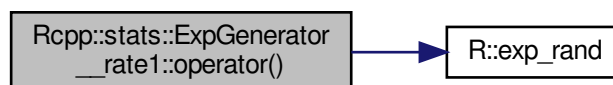
### 6.211.3.1 operator>()

```
double Rcpp::stats::ExpGenerator__rate1::operator() ( ) const [inline]
```

Definition at line 44 of file rexp.h.

References [R::exp\\_rand\(\)](#).

Here is the call graph for this function:



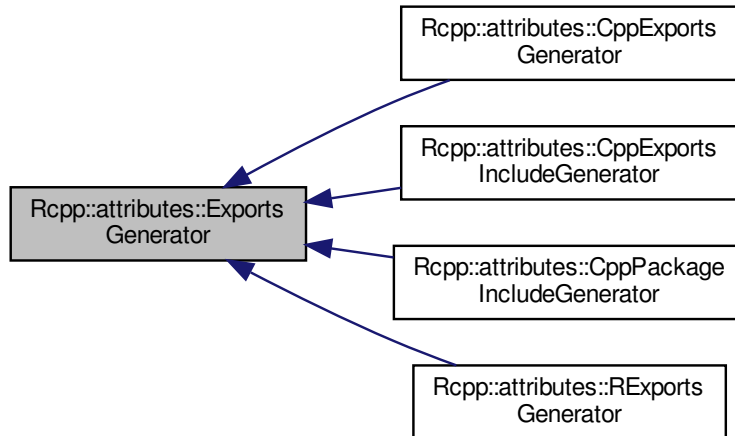
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rexp.h](#)



## 6.212 Rcpp::attributes::ExportsGenerator Class Reference

Inheritance diagram for Rcpp::attributes::ExportsGenerator:



### Public Member Functions

- virtual [~ExportsGenerator](#) ()
- const std::string & [targetFile](#) () const
- const std::string & [package](#) () const
- const std::string & [packageCpp](#) () const
- const std::string [packageCppPrefix](#) () const
- virtual void [writeBegin](#) ()=0
- void [writeFunctions](#) (const [SourceFileAttributes](#) &attributes, bool verbose)
- virtual void [writeEnd](#) (bool hasPackageInit)=0
- virtual bool [commit](#) (const std::vector< std::string > &includes)=0
- bool [remove](#) ()
- [operator std::ostream &](#) ()

### Protected Member Functions

- [ExportsGenerator](#) (const std::string &[targetFile](#), const std::string &[package](#), const std::string &commentPrefix)
- std::ostream & [ostr](#) ()
- bool [hasCplusplusInterface](#) () const
- std::string [exportValidationFunction](#) ()
- std::string [exportValidationFunctionRegisteredName](#) ()
- std::string [registerCCallableExportedName](#) ()
- bool [commit](#) (const std::string &preamble=std::string())
- std::string [dotNameHelper](#) (const std::string &name) const

## Private Member Functions

- [ExportsGenerator](#) (const [ExportsGenerator](#) &)
- [ExportsGenerator](#) & operator= (const [ExportsGenerator](#) &)
- virtual void [doWriteFunctions](#) (const [SourceFileAttributes](#) &attributes, bool verbose)=0
- bool [isSafeToOverwrite](#) () const
- std::string [generatorToken](#) () const

## Private Attributes

- std::string [targetFile\\_](#)
- std::string [package\\_](#)
- std::string [packageCpp\\_](#)
- std::string [commentPrefix\\_](#)
- std::string [existingCode\\_](#)
- std::ostream [codeStream\\_](#)
- bool [hasCppInterface\\_](#)

### 6.212.1 Detailed Description

Definition at line 599 of file attributes.cpp.

### 6.212.2 Constructor & Destructor Documentation

#### 6.212.2.1 ExportsGenerator() [1/2]

```
Rcpp::attributes::ExportsGenerator::ExportsGenerator (
    const std::string & targetFile,
    const std::string & package,
    const std::string & commentPrefix ) [protected]
```

Definition at line 1866 of file attributes.cpp.

References [existingCode\\_](#), [Rcpp::exists\(\)](#), [isSafeToOverwrite\(\)](#), [packageCpp\\_](#), and [targetFile\\_](#).

Here is the call graph for this function:



### 6.212.2.2 ExportsGenerator() [2/2]

```
Rcpp::attributes::ExportsGenerator::ExportsGenerator (  
    const ExportsGenerator & ) [private]
```

### 6.212.2.3 ~ExportsGenerator()

```
virtual Rcpp::attributes::ExportsGenerator::~~ExportsGenerator ( ) [inline], [virtual]
```

Definition at line 611 of file attributes.cpp.

## 6.212.3 Member Function Documentation

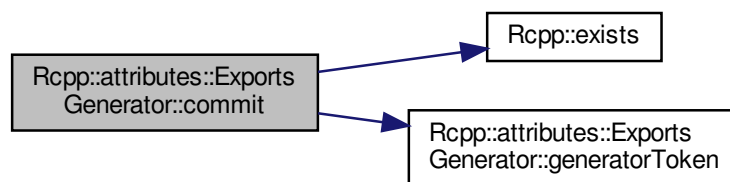
### 6.212.3.1 commit() [1/2]

```
bool Rcpp::attributes::ExportsGenerator::commit (  
    const std::string & preamble = std::string() ) [protected]
```

Definition at line 1905 of file attributes.cpp.

References `codeStream_`, `commentPrefix_`, `existingCode_`, `Rcpp::exists()`, `generatorToken()`, and `targetFile_`.

Here is the call graph for this function:



### 6.212.3.2 commit() [2/2]

```
virtual bool Rcpp::attributes::ExportsGenerator::commit (
    const std::vector< std::string > & includes ) [pure virtual]
```

Implemented in [Rcpp::attributes::REExportsGenerator](#), [Rcpp::attributes::CppPackageIncludeGenerator](#), [Rcpp::attributes::CppExportsIncludeGenerator](#), and [Rcpp::attributes::CppExportsGenerator](#).

Referenced by [Rcpp::attributes::CppExportsGenerator::commit\(\)](#), [Rcpp::attributes::CppExportsIncludeGenerator::commit\(\)](#), [Rcpp::attributes::CppPackageIncludeGenerator::commit\(\)](#), and [Rcpp::attributes::REExportsGenerator::commit\(\)](#).

### 6.212.3.3 dotNameHelper()

```
std::string Rcpp::attributes::ExportsGenerator::dotNameHelper (
    const std::string & name ) const [protected]
```

Definition at line 1950 of file `attributes.cpp`.

### 6.212.3.4 doWriteFunctions()

```
virtual void Rcpp::attributes::ExportsGenerator::doWriteFunctions (
    const SourceFileAttributes & attributes,
    bool verbose ) [private], [pure virtual]
```

Implemented in [Rcpp::attributes::REExportsGenerator](#), [Rcpp::attributes::CppExportsIncludeGenerator](#), [Rcpp::attributes::CppExportsGenerator](#), and [Rcpp::attributes::CppPackageIncludeGenerator](#).

Referenced by `writeFunctions()`.

### 6.212.3.5 exportValidationFunction()

```
std::string Rcpp::attributes::ExportsGenerator::exportValidationFunction ( ) [inline], [protected]
```

Definition at line 647 of file `attributes.cpp`.

Referenced by `exportValidationFunctionRegisteredName()`, and [Rcpp::attributes::CppExportsGenerator::writeEnd\(\)](#).

### 6.212.3.6 exportValidationFunctionRegisteredName()

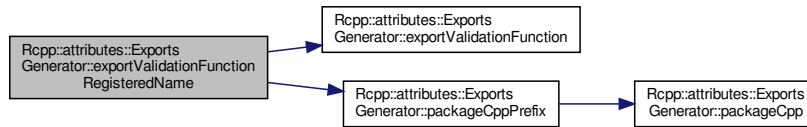
```
std::string Rcpp::attributes::ExportsGenerator::exportValidationFunctionRegisteredName ( ) [inline],
[protected]
```

Definition at line 650 of file attributes.cpp.

References `exportValidationFunction()`, and `packageCppPrefix()`.

Referenced by `Rcpp::attributes::CppExportsIncludeGenerator::writeBegin()`, and `Rcpp::attributes::CppExportsGenerator::writeEnd()`.

Here is the call graph for this function:



### 6.212.3.7 generatorToken()

```
std::string Rcpp::attributes::ExportsGenerator::generatorToken ( ) const [inline], [private]
```

Definition at line 681 of file attributes.cpp.

Referenced by `commit()`, and `isSafeToOverwrite()`.

### 6.212.3.8 hasCppInterface()

```
bool Rcpp::attributes::ExportsGenerator::hasCppInterface ( ) const [inline], [protected]
```

Definition at line 642 of file attributes.cpp.

References `hasCppInterface_`.

Referenced by `Rcpp::attributes::CppExportsGenerator::commit()`, `Rcpp::attributes::CppExportsIncludeGenerator::commit()`, `Rcpp::attributes::CppPackageIncludeGenerator::commit()`, `Rcpp::attributes::CppExportsGenerator::writeEnd()`, `Rcpp::attributes::CppPackageIncludeGenerator::writeEnd()`, and `Rcpp::attributes::REExportsGenerator::writeEnd()`.

### 6.212.3.9 isSafeToOverwrite()

```
bool Rcpp::attributes::ExportsGenerator::isSafeToOverwrite ( ) const [inline], [private]
```

Definition at line 674 of file attributes.cpp.

References existingCode\_, and generatorToken().

Referenced by ExportsGenerator().

Here is the call graph for this function:



### 6.212.3.10 operator std::ostream &()

```
Rcpp::attributes::ExportsGenerator::operator std::ostream & ( ) [inline]
```

Definition at line 631 of file attributes.cpp.

References codeStream\_.

### 6.212.3.11 operator=()

```
ExportsGenerator& Rcpp::attributes::ExportsGenerator::operator= (
    const ExportsGenerator & ) [private]
```

### 6.212.3.12 ostr()

```
std::ostream& Rcpp::attributes::ExportsGenerator::ostr ( ) [inline], [protected]
```

Definition at line 638 of file attributes.cpp.

References codeStream\_.

Referenced by Rcpp::attributes::CppExportsGenerator::commit(), Rcpp::attributes::CppExportsIncludeGenerator::commit(), Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions(), Rcpp::attributes::REExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::getCCallable(), Rcpp::attributes::CppExportsGenerator::registerCCallable(), Rcpp::attributes::CppExportsIncludeGenerator::writeBegin(), Rcpp::attributes::CppExportsGenerator::writeEnd(), Rcpp::attributes::CppExportsIncludeGenerator::writeEnd(), Rcpp::attributes::CppPackageIncludeGenerator::writeEnd(), and Rcpp::attributes::REExportsGenerator::writeEnd().

### 6.212.3.13 package()

```
const std::string& Rcpp::attributes::ExportsGenerator::package ( ) const [inline]
```

Definition at line 615 of file attributes.cpp.

References package\_.

Referenced by Rcpp::attributes::REExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::getCCallable(), Rcpp::attributes::CppExportsGenerator::registerCCallable(), Rcpp::attributes::CppExportsIncludeGenerator::writeBegin(), and Rcpp::attributes::REExportsGenerator::writeEnd().

### 6.212.3.14 packageCpp()

```
const std::string& Rcpp::attributes::ExportsGenerator::packageCpp ( ) const [inline]
```

Definition at line 616 of file attributes.cpp.

References packageCpp\_.

Referenced by Rcpp::attributes::CppExportsIncludeGenerator::commit(), Rcpp::attributes::CppExportsIncludeGenerator::getHeaderGuard(), Rcpp::attributes::CppPackageIncludeGenerator::getHeaderGuard(), packageCppPrefix(), Rcpp::attributes::CppExportsIncludeGenerator::writeBegin(), Rcpp::attributes::CppExportsGenerator::writeEnd(), and Rcpp::attributes::CppPackageIncludeGenerator::writeEnd().

### 6.212.3.15 packageCppPrefix()

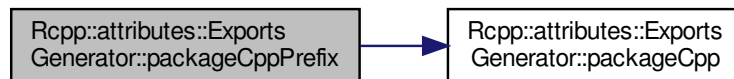
```
const std::string Rcpp::attributes::ExportsGenerator::packageCppPrefix ( ) const [inline]
```

Definition at line 617 of file attributes.cpp.

References packageCpp().

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions(), Rcpp::attributes::REExportsGenerator::doWriteFunctions(), exportValidationFunctionRegisteredName(), Rcpp::attributes::CppExportsGenerator::registerCallable(), registerCallableExportedName(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

Here is the call graph for this function:



### 6.212.3.16 registerCallableExportedName()

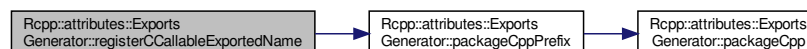
```
std::string Rcpp::attributes::ExportsGenerator::registerCallableExportedName ( ) [inline], [protected]
```

Definition at line 653 of file attributes.cpp.

References packageCppPrefix().

Referenced by Rcpp::attributes::CppExportsGenerator::writeEnd(), and Rcpp::attributes::REExportsGenerator::writeEnd().

Here is the call graph for this function:





**6.212.3.17 remove()**

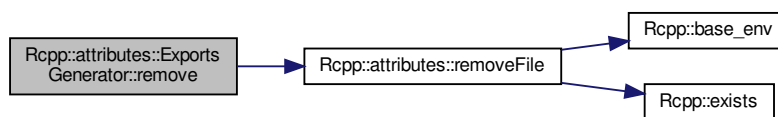
```
bool Rcpp::attributes::ExportsGenerator::remove ( )
```

Definition at line 1945 of file attributes.cpp.

References `Rcpp::attributes::removeFile()`, and `targetFile_`.

Referenced by `Rcpp::attributes::CppExportsIncludeGenerator::commit()`, and `Rcpp::attributes::CppPackageIncludeGenerator::commit()`.

Here is the call graph for this function:

**6.212.3.18 targetFile()**

```
const std::string& Rcpp::attributes::ExportsGenerator::targetFile ( ) const [inline]
```

Definition at line 614 of file attributes.cpp.

References `targetFile_`.

Referenced by `Rcpp::attributes::CppExportsGenerator::writeEnd()`.

**6.212.3.19 writeBegin()**

```
virtual void Rcpp::attributes::ExportsGenerator::writeBegin ( ) [pure virtual]
```

Implemented in [Rcpp::attributes::REExportsGenerator](#), [Rcpp::attributes::CppPackageIncludeGenerator](#), [Rcpp::attributes::CppExportsIncludeGenerator](#), and [Rcpp::attributes::CppExportsGenerator](#).

### 6.212.3.20 writeEnd()

```
virtual void Rcpp::attributes::ExportsGenerator::writeEnd (
    bool hasPackageInit ) [pure virtual]
```

Implemented in [Rcpp::attributes::REExportsGenerator](#), [Rcpp::attributes::CppPackageIncludeGenerator](#), [Rcpp::attributes::CppExportsInclud](#) and [Rcpp::attributes::CppExportsGenerator](#).

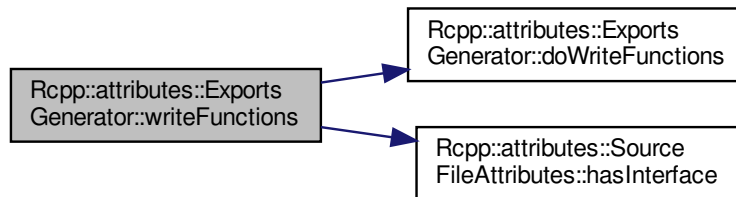
### 6.212.3.21 writeFunctions()

```
void Rcpp::attributes::ExportsGenerator::writeFunctions (
    const SourceFileAttributes & attributes,
    bool verbose )
```

Definition at line 1892 of file `attributes.cpp`.

References [doWriteFunctions\(\)](#), [hasCppInterface\\_](#), [Rcpp::attributes::SourceFileAttributes::hasInterface\(\)](#), and [Rcpp::attributes::kInterfaceCpp](#).

Here is the call graph for this function:



## 6.212.4 Member Data Documentation

### 6.212.4.1 codeStream\_

```
std::ostream Rcpp::attributes::ExportsGenerator::codeStream_ [private]
```

Definition at line 691 of file `attributes.cpp`.

Referenced by [commit\(\)](#), [operator std::ostream &\(\)](#), and [ostr\(\)](#).

#### 6.212.4.2 commentPrefix\_

```
std::string Rcpp::attributes::ExportsGenerator::commentPrefix_ [private]
```

Definition at line 689 of file attributes.cpp.

Referenced by commit().

#### 6.212.4.3 existingCode\_

```
std::string Rcpp::attributes::ExportsGenerator::existingCode_ [private]
```

Definition at line 690 of file attributes.cpp.

Referenced by commit(), ExportsGenerator(), and isSafeToOverwrite().

#### 6.212.4.4 hasCppInterface\_

```
bool Rcpp::attributes::ExportsGenerator::hasCppInterface_ [private]
```

Definition at line 692 of file attributes.cpp.

Referenced by hasCppInterface(), and writeFunctions().

#### 6.212.4.5 package\_

```
std::string Rcpp::attributes::ExportsGenerator::package_ [private]
```

Definition at line 687 of file attributes.cpp.

Referenced by package().

#### 6.212.4.6 packageCpp\_

```
std::string Rcpp::attributes::ExportsGenerator::packageCpp_ [private]
```

Definition at line 688 of file attributes.cpp.

Referenced by ExportsGenerator(), and packageCpp().

### 6.212.4.7 targetFile\_

```
std::string Rcpp::attributes::ExportsGenerator::targetFile_ [private]
```

Definition at line 686 of file attributes.cpp.

Referenced by `commit()`, `ExportsGenerator()`, `remove()`, and `targetFile()`.

The documentation for this class was generated from the following file:

- [src/attributes.cpp](#)

## 6.213 Rcpp::attributes::ExportsGenerators Class Reference

### Public Types

- typedef `std::vector< ExportsGenerator * >::iterator` `ltr`

### Public Member Functions

- [ExportsGenerators](#) ()
- virtual `~ExportsGenerators` ()
- void `add` ([ExportsGenerator](#) \*pGenerator)
- void `writeBegin` ()
- void `writeFunctions` (const [SourceFileAttributes](#) &attributes, bool verbose)
- void `writeEnd` (bool hasPackageInit)
- `std::vector< std::string >` `commit` (const `std::vector< std::string >` &includes)
- `std::vector< std::string >` `remove` ()

### Private Member Functions

- [ExportsGenerators](#) (const [ExportsGenerators](#) &)
- [ExportsGenerators](#) & `operator=` (const [ExportsGenerators](#) &)

### Private Attributes

- `std::vector< ExportsGenerator * >` `generators_`

### 6.213.1 Detailed Description

Definition at line 790 of file attributes.cpp.

## 6.213.2 Member Typedef Documentation

### 6.213.2.1 Itr

```
typedef std::vector<ExportsGenerator*>::iterator Rcpp::attributes::ExportsGenerators::Itr
```

Definition at line 792 of file attributes.cpp.

## 6.213.3 Constructor & Destructor Documentation

### 6.213.3.1 ExportsGenerators() [1/2]

```
Rcpp::attributes::ExportsGenerators::ExportsGenerators ( ) [inline]
```

Definition at line 794 of file attributes.cpp.

### 6.213.3.2 ~ExportsGenerators()

```
Rcpp::attributes::ExportsGenerators::~ExportsGenerators ( ) [virtual]
```

Definition at line 2553 of file attributes.cpp.

References generators\_.

### 6.213.3.3 ExportsGenerators() [2/2]

```
Rcpp::attributes::ExportsGenerators::ExportsGenerators (
    const ExportsGenerators & ) [private]
```

## 6.213.4 Member Function Documentation

#### 6.213.4.1 add()

```
void Rcpp::attributes::ExportsGenerators::add (  
    ExportsGenerator * pGenerator )
```

Definition at line 2562 of file attributes.cpp.

References generators\_.

Referenced by compileAttributes().

#### 6.213.4.2 commit()

```
std::vector< std::string > Rcpp::attributes::ExportsGenerators::commit (  
    const std::vector< std::string > & includes )
```

Definition at line 2584 of file attributes.cpp.

References generators\_.

Referenced by compileAttributes().

#### 6.213.4.3 operator=()

```
ExportsGenerators& Rcpp::attributes::ExportsGenerators::operator= (  
    const ExportsGenerators & ) [private]
```

#### 6.213.4.4 remove()

```
std::vector< std::string > Rcpp::attributes::ExportsGenerators::remove ( )
```

Definition at line 2598 of file attributes.cpp.

References generators\_.

Referenced by compileAttributes().

#### 6.213.4.5 writeBegin()

```
void Rcpp::attributes::ExportsGenerators::writeBegin ( )
```

Definition at line 2566 of file attributes.cpp.

References [generators\\_](#).

Referenced by [compileAttributes\(\)](#).

#### 6.213.4.6 writeEnd()

```
void Rcpp::attributes::ExportsGenerators::writeEnd (
    bool hasPackageInit )
```

Definition at line 2578 of file attributes.cpp.

References [generators\\_](#).

Referenced by [compileAttributes\(\)](#).

#### 6.213.4.7 writeFunctions()

```
void Rcpp::attributes::ExportsGenerators::writeFunctions (
    const SourceFileAttributes & attributes,
    bool verbose )
```

Definition at line 2571 of file attributes.cpp.

References [generators\\_](#).

Referenced by [compileAttributes\(\)](#).

### 6.213.5 Member Data Documentation

#### 6.213.5.1 generators\_

```
std::vector<ExportsGenerator\*> Rcpp::attributes::ExportsGenerators::generators_ [private]
```

Definition at line 817 of file attributes.cpp.

Referenced by [add\(\)](#), [commit\(\)](#), [remove\(\)](#), [writeBegin\(\)](#), [writeEnd\(\)](#), [writeFunctions\(\)](#), and [~ExportsGenerators\(\)](#).

The documentation for this class was generated from the following file:

- [src/attributes.cpp](#)

## 6.214 Rcpp::traits::Extractor< RTYPE, NA, VECTOR > Struct Template Reference

```
#include <Extractor.h>
```

### Public Types

- typedef VECTOR [type](#)

#### 6.214.1 Detailed Description

```
template<int RTYPE, bool NA, typename VECTOR>  
struct Rcpp::traits::Extractor< RTYPE, NA, VECTOR >
```

Definition at line 29 of file Extractor.h.

#### 6.214.2 Member Typedef Documentation

##### 6.214.2.1 type

```
template<int RTYPE, bool NA, typename VECTOR >  
typedef VECTOR Rcpp::traits::Extractor< RTYPE, NA, VECTOR >::type
```

Definition at line 30 of file Extractor.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/[Extractor.h](#)

## 6.215 tinyformat::detail::is\_convertible< T1, T2 >::fail Struct Reference

### Public Attributes

- char [dummy](#) [2]



### 6.215.1 Detailed Description

```
template<typename T1, typename T2>
struct tinyformat::detail::is_convertible< T1, T2 >::fail
```

Definition at line 180 of file tinyformat.h.

### 6.215.2 Member Data Documentation

#### 6.215.2.1 dummy

```
template<typename T1 , typename T2 >
char tinyformat::detail::is_convertible< T1, T2 >::fail::dummy[2]
```

Definition at line 180 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/utis/tinyformat/tinyformat.h](#)

## 6.216 Rcpp::Fast< VECTOR > Class Template Reference

```
#include <Fast.h>
```

### Public Types

- typedef VECTOR::stored\_type [value\\_type](#)

### Public Member Functions

- [Fast](#) (VECTOR &v\_)
- [value\\_type](#) & [operator\[\]](#) (R\_xlen\_t i)
- const [value\\_type](#) & [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

### Private Attributes

- VECTOR & [v](#)
- [value\\_type](#) \* [data](#)

### 6.216.1 Detailed Description

```
template<typename VECTOR>
class Rcpp::Fast< VECTOR >
```

Definition at line 27 of file Fast.h.

### 6.216.2 Member Typedef Documentation

#### 6.216.2.1 value\_type

```
template<typename VECTOR >
typedef VECTOR::stored_type Rcpp::Fast< VECTOR >::value_type
```

Definition at line 29 of file Fast.h.

### 6.216.3 Constructor & Destructor Documentation

#### 6.216.3.1 Fast()

```
template<typename VECTOR >
Rcpp::Fast< VECTOR >::Fast (
    VECTOR & v_ ) [inline]
```

Definition at line 31 of file Fast.h.

### 6.216.4 Member Function Documentation

#### 6.216.4.1 operator[]() [1/2]

```
template<typename VECTOR >
value_type& Rcpp::Fast< VECTOR >::operator[] (
    R_xlen_t i ) [inline]
```

Definition at line 33 of file Fast.h.

References Rcpp::Fast< VECTOR >::data.

### 6.216.4.2 operator[]() [2/2]

```
template<typename VECTOR >
const value_type& Rcpp::Fast< VECTOR >::operator[] (
    R_xlen_t i) const [inline]
```

Definition at line 34 of file Fast.h.

References Rcpp::Fast< VECTOR >::data.

### 6.216.4.3 size()

```
template<typename VECTOR >
R_xlen_t Rcpp::Fast< VECTOR >::size ( ) const [inline]
```

Definition at line 35 of file Fast.h.

References Rcpp::Fast< VECTOR >::v.

## 6.216.5 Member Data Documentation

### 6.216.5.1 data

```
template<typename VECTOR >
value_type* Rcpp::Fast< VECTOR >::data [private]
```

Definition at line 39 of file Fast.h.

Referenced by Rcpp::Fast< VECTOR >::operator[]().

### 6.216.5.2 v

```
template<typename VECTOR >
VECTOR& Rcpp::Fast< VECTOR >::v [private]
```

Definition at line 38 of file Fast.h.

Referenced by Rcpp::Fast< VECTOR >::size().

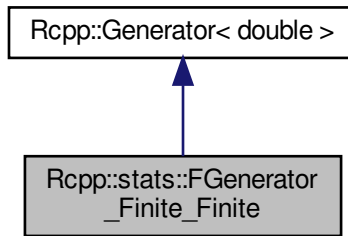
The documentation for this class was generated from the following file:

- inst/include/Rcpp/[Fast.h](#)

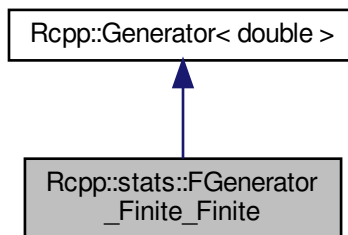
## 6.217 Rcpp::stats::FGenerator\_Finite\_Finite Class Reference

```
#include <rf.h>
```

Inheritance diagram for Rcpp::stats::FGenerator\_Finite\_Finite:



Collaboration diagram for Rcpp::stats::FGenerator\_Finite\_Finite:



### Public Member Functions

- [FGenerator\\_Finite\\_Finite](#) (double n1\_, double n2\_)
- double [operator\(\)](#) () const

### Private Attributes

- double [n1\\_\\_2](#)
- double [n2\\_\\_2](#)
- double [ratio](#)

## Additional Inherited Members

### 6.217.1 Detailed Description

Definition at line 28 of file rf.h.

### 6.217.2 Constructor & Destructor Documentation

#### 6.217.2.1 FGenerator\_Finite\_Finite()

```
Rcpp::stats::FGenerator_Finite_Finite::FGenerator_Finite_Finite (  
    double n1_,  
    double n2_ ) [inline]
```

Definition at line 31 of file rf.h.

### 6.217.3 Member Function Documentation

#### 6.217.3.1 operator>()()

```
double Rcpp::stats::FGenerator_Finite_Finite::operator() ( ) const [inline]
```

Definition at line 34 of file rf.h.

References `n1__2`, `n2__2`, and `ratio`.

### 6.217.4 Member Data Documentation

#### 6.217.4.1 n1\_\_2

```
double Rcpp::stats::FGenerator_Finite_Finite::n1__2 [private]
```

Definition at line 41 of file rf.h.

Referenced by `operator>()()`.

### 6.217.4.2 n2\_\_2

```
double Rcpp::stats::FGenerator_Finite_Finite::n2__2 [private]
```

Definition at line 41 of file rf.h.

Referenced by operator(()).

### 6.217.4.3 ratio

```
double Rcpp::stats::FGenerator_Finite_Finite::ratio [private]
```

Definition at line 41 of file rf.h.

Referenced by operator(()).

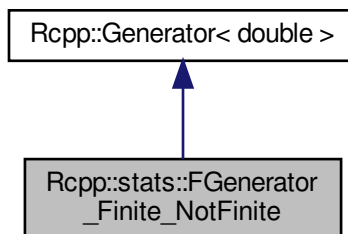
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rf.h](#)

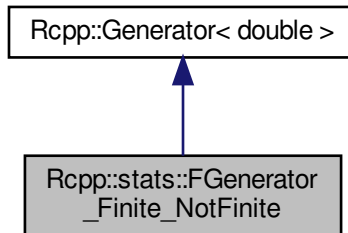
## 6.218 Rcpp::stats::FGenerator\_Finite\_NotFinite Class Reference

```
#include <rf.h>
```

Inheritance diagram for Rcpp::stats::FGenerator\_Finite\_NotFinite:



Collaboration diagram for Rcpp::stats::FGenerator\_Finite\_NotFinite:



## Public Member Functions

- `FGenerator_Finite_NotFinite` (double `n1_`)
- double `operator()` () const

## Private Attributes

- double `n1`
- double `n1__2`

## Additional Inherited Members

### 6.218.1 Detailed Description

Definition at line 58 of file `rf.h`.

### 6.218.2 Constructor & Destructor Documentation

#### 6.218.2.1 FGenerator\_Finite\_NotFinite()

```
Rcpp::stats::FGenerator_Finite_NotFinite::FGenerator_Finite_NotFinite (  
    double n1_ ) [inline]
```

Definition at line 61 of file `rf.h`.

### 6.218.3 Member Function Documentation

#### 6.218.3.1 operator()

```
double Rcpp::stats::FGenerator_Finite_NotFinite::operator() ( ) const [inline]
```

Definition at line 63 of file rf.h.

References `n1`, and `n1__2`.

### 6.218.4 Member Data Documentation

#### 6.218.4.1 n1

```
double Rcpp::stats::FGenerator_Finite_NotFinite::n1 [private]
```

Definition at line 69 of file rf.h.

Referenced by `operator()`.

#### 6.218.4.2 n1\_\_2

```
double Rcpp::stats::FGenerator_Finite_NotFinite::n1__2 [private]
```

Definition at line 69 of file rf.h.

Referenced by `operator()`.

The documentation for this class was generated from the following file:

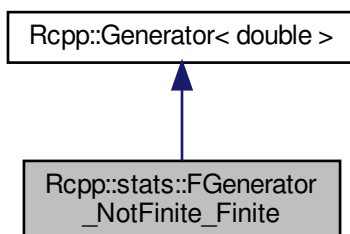
- [inst/include/Rcpp/stats/random/rf.h](#)



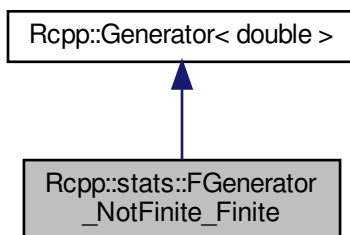
## 6.219 Rcpp::stats::FGenerator\_NotFinite\_Finite Class Reference

```
#include <rf.h>
```

Inheritance diagram for Rcpp::stats::FGenerator\_NotFinite\_Finite:



Collaboration diagram for Rcpp::stats::FGenerator\_NotFinite\_Finite:



### Public Member Functions

- [FGenerator\\_NotFinite\\_Finite](#) (double n2\_)
- double [operator\(\)](#) () const

### Private Attributes

- double [n2](#)
- double [n2\\_\\_2](#)

## Additional Inherited Members

### 6.219.1 Detailed Description

Definition at line 44 of file rf.h.

### 6.219.2 Constructor & Destructor Documentation

#### 6.219.2.1 FGenerator\_NotFinite\_Finite()

```
Rcpp::stats::FGenerator_NotFinite_Finite::FGenerator_NotFinite_Finite (  
    double n2_ ) [inline]
```

Definition at line 47 of file rf.h.

### 6.219.3 Member Function Documentation

#### 6.219.3.1 operator>()

```
double Rcpp::stats::FGenerator_NotFinite_Finite::operator() ( ) const [inline]
```

Definition at line 49 of file rf.h.

References n2, and n2\_\_2.

### 6.219.4 Member Data Documentation

#### 6.219.4.1 n2

```
double Rcpp::stats::FGenerator_NotFinite_Finite::n2 [private]
```

Definition at line 55 of file rf.h.

Referenced by operator()).

### 6.219.4.2 n2\_\_2

```
double Rcpp::stats::FGenerator_NotFinite_Finite::n2__2 [private]
```

Definition at line 55 of file rf.h.

Referenced by operator()).

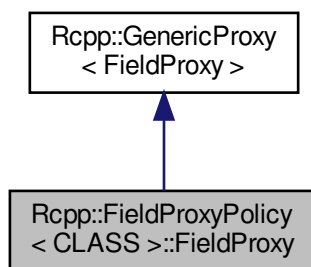
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rf.h

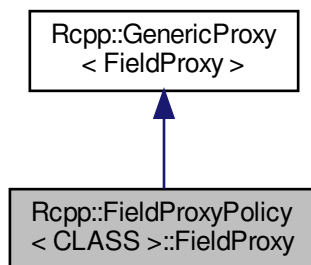
## 6.220 Rcpp::FieldProxyPolicy< CLASS >::FieldProxy Class Reference

```
#include <FieldProxy.h>
```

Inheritance diagram for Rcpp::FieldProxyPolicy< CLASS >::FieldProxy:



Collaboration diagram for Rcpp::FieldProxyPolicy< CLASS >::FieldProxy:



## Public Member Functions

- [FieldProxy](#) (CLASS &v, const std::string &name)
- [FieldProxy](#) & [operator=](#) (const [FieldProxy](#) &rhs)
- template<typename T >  
[FieldProxy](#) & [operator=](#) (const T &rhs)
- template<typename T >  
[operator T](#) () const
- [operator SEXP](#) () const
- template<typename CLASS >  
[FieldProxyPolicy](#)< CLASS >::[FieldProxy](#) & [operator=](#) (const [FieldProxyPolicy](#)< CLASS >::[FieldProxy](#) &rhs)
- template<typename T >  
[FieldProxyPolicy](#)< CLASS >::[FieldProxy](#) & [operator=](#) (const T &rhs)

## Private Member Functions

- SEXP [get](#) () const
- void [set](#) (SEXP x)

## Private Attributes

- CLASS & [parent](#)
- const std::string & [field\\_name](#)

### 6.220.1 Detailed Description

```
template<typename CLASS>
class Rcpp::FieldProxyPolicy< CLASS >::FieldProxy
```

Definition at line 27 of file FieldProxy.h.

### 6.220.2 Constructor & Destructor Documentation

#### 6.220.2.1 FieldProxy()

```
template<typename CLASS >
Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::FieldProxy (
    CLASS & v,
    const std::string & name ) [inline]
```

Definition at line 29 of file FieldProxy.h.

## 6.220.3 Member Function Documentation

### 6.220.3.1 get()

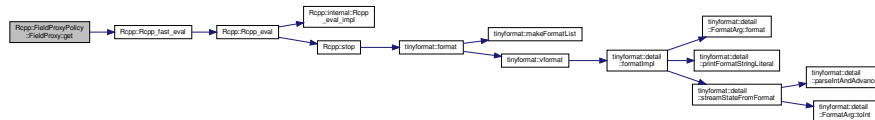
```
template<typename CLASS >
SEXP Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get ( ) const [inline], [private]
```

Definition at line 43 of file FieldProxy.h.

References Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::field\_name, Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::parent, and Rcpp::Rcpp\_fast\_eval().

Referenced by Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator SEXP(), and Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator=().

Here is the call graph for this function:



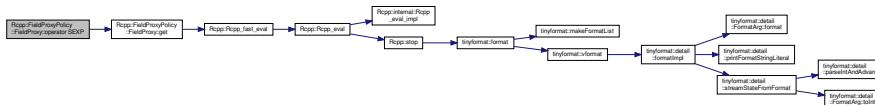
### 6.220.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator SEXP ( ) const [inline]
```

Definition at line 37 of file FieldProxy.h.

References Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get().

Here is the call graph for this function:



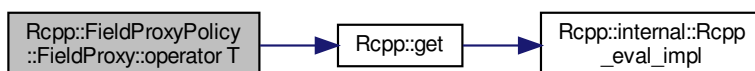
### 6.220.3.3 operator T()

```
template<typename CLASS >
template<typename T >
Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator T
```

Definition at line 193 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



### 6.220.3.4 operator=() [1/4]

```
template<typename CLASS >
FieldProxy& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator= (
    const FieldProxy & rhs )
```

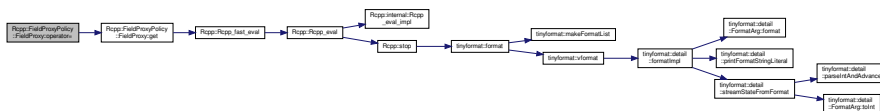
### 6.220.3.5 operator=() [2/4]

```
template<typename CLASS >
template<typename CLASS >
FieldProxyPolicy<CLASS>::FieldProxy& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator= (
    const FieldProxyPolicy< CLASS >::FieldProxy & rhs )
```

Definition at line 178 of file proxy.h.

References Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get().

Here is the call graph for this function:



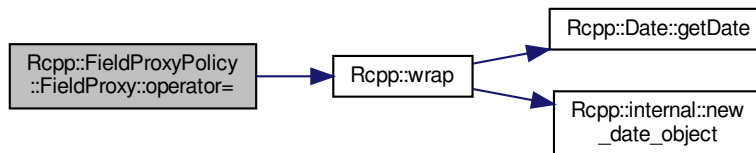
### 6.220.3.6 operator=() [3/4]

```
template<typename CLASS >
template<typename T >
FieldProxyPolicy<CLASS>::FieldProxy& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator= (
    const T & rhs )
```

Definition at line 186 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:



### 6.220.3.7 operator=() [4/4]

```
template<typename CLASS >
template<typename T >
FieldProxy& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::operator= (
    const T & rhs )
```

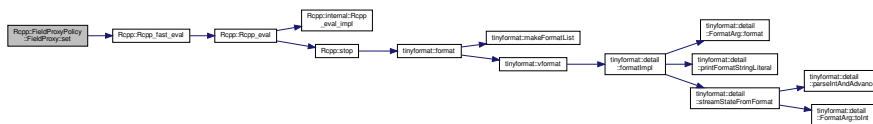
### 6.220.3.8 set()

```
template<typename CLASS >
void Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::set (
    SEXP x ) [inline], [private]
```

Definition at line 48 of file FieldProxy.h.

References Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::field\_name, Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::parent, and Rcpp::Rcpp\_fast\_eval().

Here is the call graph for this function:



## 6.220.4 Member Data Documentation

### 6.220.4.1 field\_name

```
template<typename CLASS >
const std::string& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::field_name [private]
```

Definition at line 41 of file FieldProxy.h.

Referenced by `Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get()`, and `Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::set()`.

### 6.220.4.2 parent

```
template<typename CLASS >
CLASS& Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::parent [private]
```

Definition at line 40 of file FieldProxy.h.

Referenced by `Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::get()`, and `Rcpp::FieldProxyPolicy< CLASS >::FieldProxy::set()`.

The documentation for this class was generated from the following files:

- `inst/include/Rcpp/proxy/FieldProxy.h`
- `inst/include/Rcpp/api/meat/proxy.h`

## 6.221 Rcpp::FieldProxyPolicy< CLASS > Class Template Reference

```
#include <FieldProxy.h>
```

### Classes

- class `const_FieldProxy`
- class `FieldProxy`

### Public Member Functions

- `FieldProxy field` (const std::string &name)
- `const_FieldProxy field` (const std::string &name) const



## 6.221.1 Detailed Description

```
template<typename CLASS>  
class Rcpp::FieldProxyPolicy< CLASS >
```

Definition at line 24 of file FieldProxy.h.

## 6.221.2 Member Function Documentation

### 6.221.2.1 field() [1/2]

```
template<typename CLASS >  
FieldProxy Rcpp::FieldProxyPolicy< CLASS >::field (  
    const std::string & name ) [inline]
```

Definition at line 77 of file FieldProxy.h.

### 6.221.2.2 field() [2/2]

```
template<typename CLASS >  
const_FieldProxy Rcpp::FieldProxyPolicy< CLASS >::field (  
    const std::string & name ) const [inline]
```

Definition at line 80 of file FieldProxy.h.

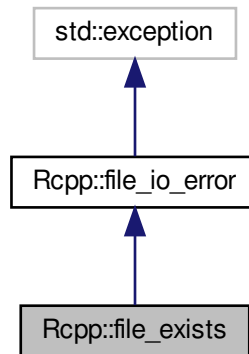
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/[FieldProxy.h](#)

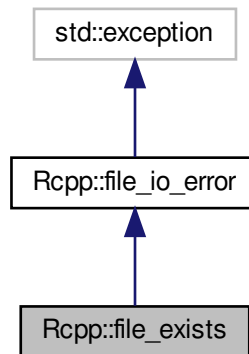
## 6.222 Rcpp::file\_exists Class Reference

```
#include <exceptions.h>
```

Inheritance diagram for Rcpp::file\_exists:



Collaboration diagram for Rcpp::file\_exists:



### Public Member Functions

- [file\\_exists](#) (const std::string &file) throw ()

### 6.222.1 Detailed Description

Definition at line 104 of file exceptions.h.

### 6.222.2 Constructor & Destructor Documentation

#### 6.222.2.1 file\_exists()

```
Rcpp::file_exists::file_exists (
    const std::string & file ) throw ( ) [inline]
```

Definition at line 106 of file exceptions.h.

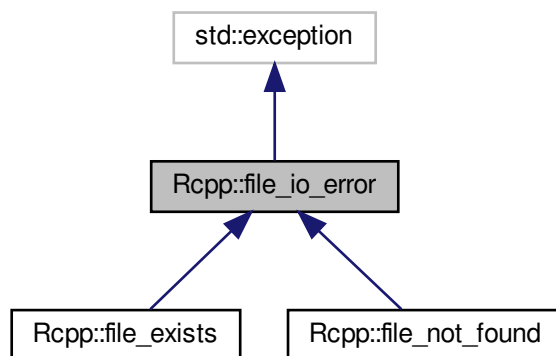
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/exceptions.h](#)

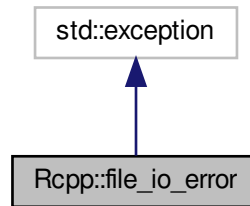
## 6.223 Rcpp::file\_io\_error Class Reference

```
#include <exceptions.h>
```

Inheritance diagram for Rcpp::file\_io\_error:



Collaboration diagram for Rcpp::file\_io\_error:



## Public Member Functions

- [file\\_io\\_error](#) (const std::string &[file](#)) throw ()
- [file\\_io\\_error](#) (int code, const std::string &[file](#)) throw ()
- [file\\_io\\_error](#) (const std::string &msg, const std::string &[file](#)) throw ()
- virtual [~file\\_io\\_error](#) () throw ()
- virtual const char \* [what](#) () const throw ()
- std::string [filePath](#) () const throw ()

## Private Attributes

- std::string [message](#)
- std::string [file](#)

### 6.223.1 Detailed Description

Definition at line 82 of file exceptions.h.

### 6.223.2 Constructor & Destructor Documentation

#### 6.223.2.1 [file\\_io\\_error](#)() [1/3]

```
Rcpp::file_io_error::file_io_error (  
    const std::string & file ) throw ( )    [inline]
```

Definition at line 84 of file exceptions.h.

### 6.223.2.2 file\_io\_error() [2/3]

```
Rcpp::file_io_error::file_io_error (
    int code,
    const std::string & file ) throw ( )    [inline]
```

Definition at line 86 of file exceptions.h.

### 6.223.2.3 file\_io\_error() [3/3]

```
Rcpp::file_io_error::file_io_error (
    const std::string & msg,
    const std::string & file ) throw ( )    [inline]
```

Definition at line 88 of file exceptions.h.

### 6.223.2.4 ~file\_io\_error()

```
virtual Rcpp::file_io_error::~~file_io_error ( ) throw ( )    [inline], [virtual]
```

Definition at line 90 of file exceptions.h.

## 6.223.3 Member Function Documentation

### 6.223.3.1 filePath()

```
std::string Rcpp::file_io_error::filePath ( ) const throw ( )    [inline]
```

Definition at line 92 of file exceptions.h.

References file.

Referenced by compileAttributes().

### 6.223.3.2 what()

```
virtual const char* Rcpp::file_io_error::what ( ) const throw ( ) [inline], [virtual]
```

Definition at line 91 of file exceptions.h.

References message.

## 6.223.4 Member Data Documentation

### 6.223.4.1 file

```
std::string Rcpp::file_io_error::file [private]
```

Definition at line 95 of file exceptions.h.

Referenced by filePath().

### 6.223.4.2 message

```
std::string Rcpp::file_io_error::message [private]
```

Definition at line 94 of file exceptions.h.

Referenced by what().

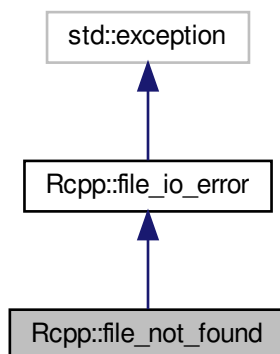
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/exceptions.h](#)

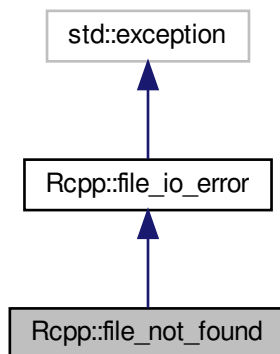
## 6.224 Rcpp::file\_not\_found Class Reference

```
#include <exceptions.h>
```

Inheritance diagram for Rcpp::file\_not\_found:



Collaboration diagram for Rcpp::file\_not\_found:



### Public Member Functions

- [file\\_not\\_found](#) (const std::string &file) throw ()

### 6.224.1 Detailed Description

Definition at line 98 of file exceptions.h.

### 6.224.2 Constructor & Destructor Documentation

#### 6.224.2.1 file\_not\_found()

```
Rcpp::file_not_found::file_not_found (
    const std::string & file ) throw ( ) [inline]
```

Definition at line 100 of file exceptions.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/exceptions.h](#)

## 6.225 Rcpp::attributes::FileInfo Class Reference

### Public Member Functions

- [FileInfo](#) (const std::string &path)
- [FileInfo](#) (const [List](#) &fileInfo)
- [List toList](#) () const
- std::string [path](#) () const
- bool [exists](#) () const
- double [lastModified](#) () const
- std::string [extension](#) () const
- bool [operator<](#) (const [FileInfo](#) &other) const
- bool [operator==](#) (const [FileInfo](#) &other) const
- bool [operator!=](#) (const [FileInfo](#) &other) const
- std::ostream & [operator<<](#) (std::ostream &os) const

### Private Attributes

- std::string [path\\_](#)
- bool [exists\\_](#)
- double [lastModified\\_](#)



## 6.225.1 Detailed Description

Definition at line 49 of file attributes.cpp.

## 6.225.2 Constructor & Destructor Documentation

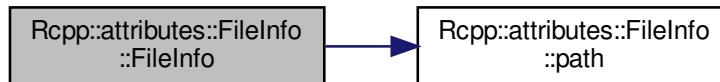
### 6.225.2.1 FileInfo() [1/2]

```
Rcpp::attributes::FileInfo::FileInfo (  
    const std::string & path ) [explicit]
```

Definition at line 2980 of file attributes.cpp.

References exists\_, lastModified\_, and path().

Here is the call graph for this function:



### 6.225.2.2 FileInfo() [2/2]

```
Rcpp::attributes::FileInfo::FileInfo (  
    const List & fileInfo ) [inline], [explicit]
```

Definition at line 56 of file attributes.cpp.

References exists\_, lastModified\_, and path\_.

## 6.225.3 Member Function Documentation

### 6.225.3.1 exists()

```
bool Rcpp::attributes::FileInfo::exists ( ) const [inline]
```

Definition at line 72 of file attributes.cpp.

References exists\_.

### 6.225.3.2 extension()

```
std::string Rcpp::attributes::FileInfo::extension ( ) const [inline]
```

Definition at line 75 of file attributes.cpp.

References path\_.

### 6.225.3.3 lastModified()

```
double Rcpp::attributes::FileInfo::lastModified ( ) const [inline]
```

Definition at line 73 of file attributes.cpp.

References lastModified\_.

### 6.225.3.4 operator"!="()

```
bool Rcpp::attributes::FileInfo::operator!= (
    const FileInfo & other ) const [inline]
```

Definition at line 93 of file attributes.cpp.

### 6.225.3.5 operator<()

```
bool Rcpp::attributes::FileInfo::operator< (
    const FileInfo & other ) const [inline]
```

Definition at line 83 of file attributes.cpp.

References path\_.

### 6.225.3.6 operator<<()

```
std::ostream& Rcpp::attributes::FileInfo::operator<< (
    std::ostream & os ) const [inline]
```

Definition at line 97 of file attributes.cpp.

References [path\\_](#).

### 6.225.3.7 operator==( )

```
bool Rcpp::attributes::FileInfo::operator==(
    const FileInfo & other ) const [inline]
```

Definition at line 87 of file attributes.cpp.

References [exists\\_](#), [lastModified\\_](#), and [path\\_](#).

### 6.225.3.8 path()

```
std::string Rcpp::attributes::FileInfo::path ( ) const [inline]
```

Definition at line 71 of file attributes.cpp.

References [path\\_](#).

Referenced by [FileInfo\(\)](#).

### 6.225.3.9 toList()

```
List Rcpp::attributes::FileInfo::toList ( ) const [inline]
```

Definition at line 63 of file attributes.cpp.

References [exists\\_](#), [lastModified\\_](#), and [path\\_](#).

## 6.225.4 Member Data Documentation

#### 6.225.4.1 exists\_

```
bool Rcpp::attributes::FileInfo::exists_ [private]
```

Definition at line 104 of file attributes.cpp.

Referenced by exists(), FileInfo(), operator==(), and toList().

#### 6.225.4.2 lastModified\_

```
double Rcpp::attributes::FileInfo::lastModified_ [private]
```

Definition at line 105 of file attributes.cpp.

Referenced by FileInfo(), lastModified(), operator==(), and toList().

#### 6.225.4.3 path\_

```
std::string Rcpp::attributes::FileInfo::path_ [private]
```

Definition at line 103 of file attributes.cpp.

Referenced by extension(), FileInfo(), operator<(), operator<<(), operator==(), path(), and toList().

The documentation for this class was generated from the following file:

- [src/attributes.cpp](#)

## 6.226 Rcpp::fixed\_call< RESULT\_TYPE > Class Template Reference

```
#include <Language.h>
```

### Public Types

- typedef RESULT\_TYPE [result\\_type](#)

### Public Member Functions

- [fixed\\_call](#) (Language call\_)
- [fixed\\_call](#) (Function fun)
- RESULT\_TYPE [operator](#)() ()

## Private Attributes

- [Language call](#)

### 6.226.1 Detailed Description

```
template<typename RESULT_TYPE = SEXP>
class Rcpp::fixed_call< RESULT_TYPE >
```

Definition at line 163 of file Language.h.

### 6.226.2 Member Typedef Documentation

#### 6.226.2.1 result\_type

```
template<typename RESULT_TYPE = SEXP>
typedef RESULT_TYPE Rcpp::fixed_call< RESULT_TYPE >::result_type
```

Definition at line 165 of file Language.h.

### 6.226.3 Constructor & Destructor Documentation

#### 6.226.3.1 fixed\_call() [1/2]

```
template<typename RESULT_TYPE = SEXP>
Rcpp::fixed_call< RESULT_TYPE >::fixed_call (
    Language call_ ) [inline]
```

Definition at line 167 of file Language.h.

#### 6.226.3.2 fixed\_call() [2/2]

```
template<typename RESULT_TYPE = SEXP>
Rcpp::fixed_call< RESULT_TYPE >::fixed_call (
    Function fun ) [inline]
```

Definition at line 168 of file Language.h.

## 6.226.4 Member Function Documentation

### 6.226.4.1 operator()

```
template<typename RESULT_TYPE = SEXP>  
RESULT_TYPE Rcpp::fixed_call< RESULT_TYPE >::operator() ( ) [inline]
```

Definition at line 170 of file Language.h.

References Rcpp::fixed\_call< RESULT\_TYPE >::call.

## 6.226.5 Member Data Documentation

### 6.226.5.1 call

```
template<typename RESULT_TYPE = SEXP>  
Language Rcpp::fixed_call< RESULT_TYPE >::call [private]
```

Definition at line 175 of file Language.h.

Referenced by Rcpp::fixed\_call< RESULT\_TYPE >::operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/[Language.h](#)

## 6.227 Rcpp::sugar::forbidden\_conversion< bool > Class Template Reference

### 6.227.1 Detailed Description

```
template<bool>  
class Rcpp::sugar::forbidden_conversion< bool >
```

Definition at line 29 of file SingleLogicalResult.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/[SingleLogicalResult.h](#)

## 6.228 Rcpp::sugar::forbidden\_conversion< true > Class Reference

```
#include <SingleLogicalResult.h>
```

### 6.228.1 Detailed Description

Definition at line 32 of file SingleLogicalResult.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/SingleLogicalResult.h

## 6.229 tinyformat::detail::FormatArg Class Reference

```
#include <tinyformat.h>
```

### Public Member Functions

- [FormatArg](#) ()
- [FormatArg](#) (const T &value)
- void [format](#) (std::ostream &out, const char \*fmtBegin, const char \*fmtEnd, int ntrunc) const
- int [toInt](#) () const

### Static Private Member Functions

- [formatImpl](#) (std::ostream &out, const char \*fmtBegin, const char \*fmtEnd, int ntrunc, const void \*value)
- [toIntImpl](#) (const void \*value)

### Private Attributes

- const void \* [m\\_value](#)
- void(\* [m\\_formatImpl](#)) (std::ostream &out, const char \*fmtBegin, const char \*fmtEnd, int ntrunc, const void \*value)
- int(\* [m\\_toIntImpl](#)) (const void \*value)

### 6.229.1 Detailed Description

Definition at line 492 of file tinyformat.h.

## 6.229.2 Constructor & Destructor Documentation

### 6.229.2.1 FormatArg() [1/2]

```
tinyformat::detail::FormatArg::FormatArg ( ) [inline]
```

Definition at line 495 of file tinyformat.h.

### 6.229.2.2 FormatArg() [2/2]

```
template<typename T >  
tinyformat::detail::FormatArg::FormatArg (  
    const T & value ) [inline]
```

Definition at line 502 of file tinyformat.h.

## 6.229.3 Member Function Documentation

### 6.229.3.1 format()

```
void tinyformat::detail::FormatArg::format (  
    std::ostream & out,  
    const char * fmtBegin,  
    const char * fmtEnd,  
    int ntrunc ) const [inline]
```

Definition at line 508 of file tinyformat.h.

References `m_formatImpl`, `m_value`, and `TINYFORMAT_ASSERT`.

Referenced by `tinyformat::detail::formatImpl()`.



### 6.229.3.2 formatImpl()

```
template<typename T >
static TINYFORMAT_HIDDEN void tinyformat::detail::FormatArg::formatImpl (
    std::ostream & out,
    const char * fmtBegin,
    const char * fmtEnd,
    int ntrunc,
    const void * value ) [inline], [static], [private]
```

Definition at line 525 of file tinyformat.h.

References tinyformat::formatValue().

Here is the call graph for this function:



### 6.229.3.3 toInt()

```
int tinyformat::detail::FormatArg::toInt ( ) const [inline]
```

Definition at line 516 of file tinyformat.h.

References m\_toIntImpl, m\_value, and TINYFORMAT\_ASSERT.

Referenced by tinyformat::detail::streamStateFromFormat().

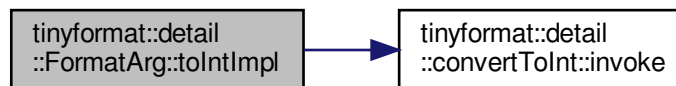
### 6.229.3.4 toIntImpl()

```
template<typename T >
static TINYFORMAT_HIDDEN int tinyformat::detail::FormatArg::toIntImpl (
    const void * value ) [inline], [static], [private]
```

Definition at line 532 of file tinyformat.h.

References tinyformat::detail::convertToInt< T, convertible >::invoke().

Here is the call graph for this function:



## 6.229.4 Member Data Documentation

### 6.229.4.1 m\_formatImpl

```
void(* tinyformat::detail::FormatArg::m_formatImpl) (std::ostream &out, const char *fmtBegin,  
const char *fmtEnd, int ntrunc, const void *value) [private]
```

Definition at line 538 of file tinyformat.h.

Referenced by format().

### 6.229.4.2 m\_toIntImpl

```
int(* tinyformat::detail::FormatArg::m_toIntImpl) (const void *value) [private]
```

Definition at line 540 of file tinyformat.h.

Referenced by toInt().

### 6.229.4.3 m\_value

```
const void* tinyformat::detail::FormatArg::m_value [private]
```

Definition at line 537 of file tinyformat.h.

Referenced by format(), and toInt().

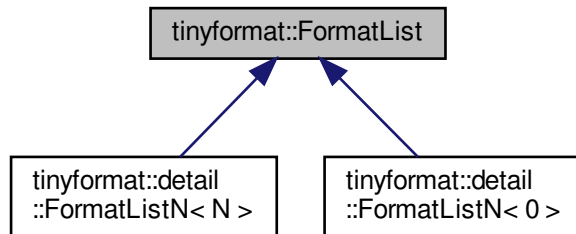
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/utils/tinyformat/tinyformat.h](#)

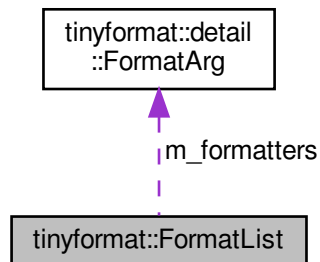
## 6.230 tinyformat::FormatList Class Reference

```
#include <tinyformat.h>
```

Inheritance diagram for tinyformat::FormatList:



Collaboration diagram for tinyformat::FormatList:



### Public Member Functions

- `FormatList` (`detail::FormatArg *formatters`, `int N`)

### Private Attributes

- `const detail::FormatArg * m_formatters`
- `int m_N`

## Friends

- void `vformat` (std::ostream &out, const char \*fmt, const `FormatList` &list)

### 6.230.1 Detailed Description

List of template arguments `format()`, held in a type-opaque way.

A const reference to `FormatList` (typedef'd as `FormatListRef`) may be conveniently used to pass arguments to non-template functions: All type information has been stripped from the arguments, leaving just enough of a common interface to perform formatting as required.

Definition at line 852 of file `tinyformat.h`.

### 6.230.2 Constructor & Destructor Documentation

#### 6.230.2.1 `FormatList()`

```
tinyformat::FormatList::FormatList (  
    detail::FormatArg * formatters,  
    int N ) [inline]
```

Definition at line 855 of file `tinyformat.h`.

### 6.230.3 Friends And Related Function Documentation

#### 6.230.3.1 `vformat`

```
void vformat (  
    std::ostream & out,  
    const char * fmt,  
    const FormatList & list ) [friend]
```

### 6.230.4 Member Data Documentation

#### 6.230.4.1 m\_formatters

```
const detail::FormatArg* tinyformat::FormatList::m_formatters [private]
```

Definition at line 862 of file tinyformat.h.

Referenced by tinyformat::vformat().

#### 6.230.4.2 m\_N

```
int tinyformat::FormatList::m_N [private]
```

Definition at line 863 of file tinyformat.h.

Referenced by tinyformat::vformat().

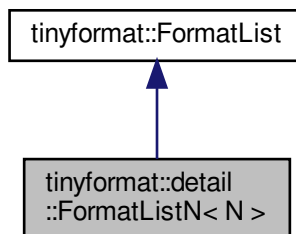
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/utils/tinyformat/tinyformat.h](#)

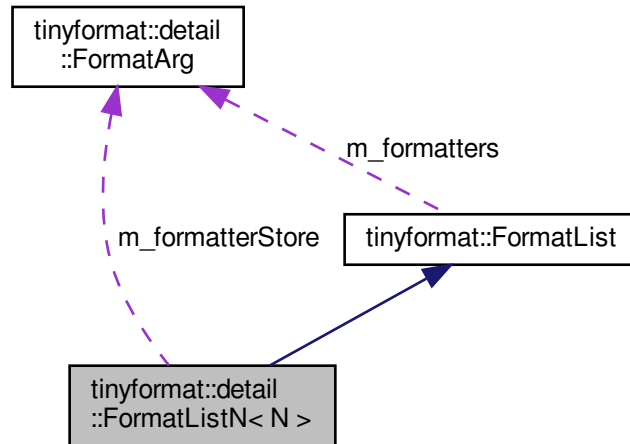
## 6.231 tinyformat::detail::FormatListN< N > Class Template Reference

```
#include <tinyformat.h>
```

Inheritance diagram for tinyformat::detail::FormatListN< N >:



Collaboration diagram for `tinyformat::detail::FormatListN< N >`:



## Public Member Functions

- void `init` (int)

## Private Attributes

- `FormatArg m_formatterStore` [N]

### 6.231.1 Detailed Description

```
template<int N>
class tinyformat::detail::FormatListN< N >
```

Definition at line 874 of file `tinyformat.h`.

### 6.231.2 Member Function Documentation

### 6.231.2.1 init()

```
template<int N>
void tinyformat::detail::FormatListN< N >::init (
    int ) [inline]
```

Definition at line 884 of file tinyformat.h.

## 6.231.3 Member Data Documentation

### 6.231.3.1 m\_formatterStore

```
template<int N>
FormatArg tinyformat::detail::FormatListN< N >::m_formatterStore[N] [private]
```

Definition at line 904 of file tinyformat.h.

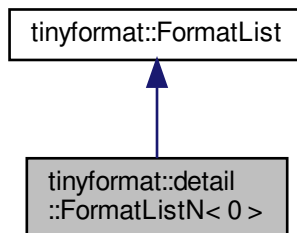
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/utils/tinyformat/tinyformat.h](#)

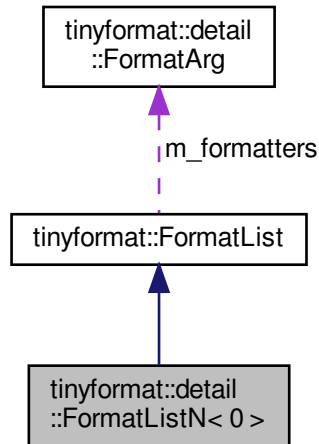
## 6.232 tinyformat::detail::FormatListN< 0 > Class Reference

```
#include <tinyformat.h>
```

Inheritance diagram for tinyformat::detail::FormatListN< 0 >:



Collaboration diagram for `tinyformat::detail::FormatListN< 0 >`:



## Public Member Functions

- [FormatListN\(\)](#)

### 6.232.1 Detailed Description

Definition at line 908 of file `tinyformat.h`.

### 6.232.2 Constructor & Destructor Documentation

#### 6.232.2.1 FormatListN()

```
tinyformat::detail::FormatListN< 0 >::FormatListN ( ) [inline]
```

Definition at line 910 of file `tinyformat.h`.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/utils/tinyformat/tinyformat.h](#)



## 6.233 `tinyformat::detail::formatValueAsType< T, fmtT, convertible >` Struct Template Reference

```
#include <tinyformat.h>
```

### Static Public Member Functions

- static void [invoke](#) (std::ostream &, const T &)

### 6.233.1 Detailed Description

```
template<typename T, typename fmtT, bool convertible = is_convertible<T, fmtT>::value>
struct tinyformat::detail::formatValueAsType< T, fmtT, convertible >
```

Definition at line 216 of file `tinyformat.h`.

### 6.233.2 Member Function Documentation

#### 6.233.2.1 `invoke()`

```
template<typename T , typename fmtT , bool convertible = is_convertible<T, fmtT>::value>
static void tinyformat::detail::formatValueAsType< T, fmtT, convertible >::invoke (
    std::ostream & ,
    const T & ) [inline], [static]
```

Definition at line 218 of file `tinyformat.h`.

References `TINYFORMAT_ASSERT`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/Utils/tinyformat/tinyformat.h](#)

## 6.234 `tinyformat::detail::formatValueAsType< T, fmtT, true >` Struct Template Reference

```
#include <tinyformat.h>
```

## Static Public Member Functions

- static void [invoke](#) (std::ostream &out, const T &value)

### 6.234.1 Detailed Description

```
template<typename T, typename fmtT>
struct tinyformat::detail::formatValueAsType< T, fmtT, true >
```

Definition at line 223 of file tinyformat.h.

### 6.234.2 Member Function Documentation

#### 6.234.2.1 invoke()

```
template<typename T , typename fmtT >
static void tinyformat::detail::formatValueAsType< T, fmtT, true >::invoke (
    std::ostream & out,
    const T & value ) [inline], [static]
```

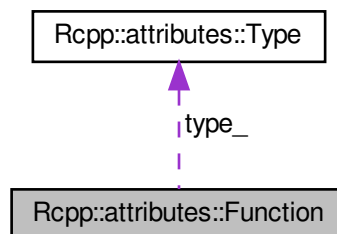
Definition at line 225 of file tinyformat.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/utils/tinyformat/[tinyformat.h](#)

## 6.235 Rcpp::attributes::Function Class Reference

Collaboration diagram for Rcpp::attributes::Function:



## Public Member Functions

- [Function](#) ()
- [Function](#) (const [Type](#) &*type*, const std::string &*name*, const std::vector< [Argument](#) > &*arguments*)
- [Function renamedTo](#) (const std::string &*name*) const
- std::string [signature](#) () const
- std::string [signature](#) (const std::string &*name*) const
- bool [isHidden](#) () const
- bool [empty](#) () const
- bool [operator==](#) (const [Function](#) &*other*) const
- bool [operator!=](#) (const [Function](#) &*other*) const
- const [Type](#) & [type](#) () const
- const std::string & [name](#) () const
- const std::vector< [Argument](#) > & [arguments](#) () const

## Private Attributes

- [Type](#) [type\\_](#)
- std::string [name\\_](#)
- std::vector< [Argument](#) > [arguments\\_](#)

### 6.235.1 Detailed Description

Definition at line 245 of file attributes.cpp.

### 6.235.2 Constructor & Destructor Documentation

#### 6.235.2.1 Function() [1/2]

```
Rcpp::attributes::Function::Function ( ) [inline]
```

Definition at line 247 of file attributes.cpp.

Referenced by [renamedTo\(\)](#).

#### 6.235.2.2 Function() [2/2]

```
Rcpp::attributes::Function::Function (
    const Type & type,
    const std::string & name,
    const std::vector< Argument > & arguments ) [inline]
```

Definition at line 248 of file attributes.cpp.

## 6.235.3 Member Function Documentation

### 6.235.3.1 arguments()

```
const std::vector<Argument>& Rcpp::attributes::Function::arguments ( ) const [inline]
```

Definition at line 280 of file attributes.cpp.

References arguments\_.

Referenced by renamedTo(), signature(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

### 6.235.3.2 empty()

```
bool Rcpp::attributes::Function::empty ( ) const [inline]
```

Definition at line 266 of file attributes.cpp.

References name().

Referenced by Rcpp::attributes::operator<<().

Here is the call graph for this function:



### 6.235.3.3 isHidden()

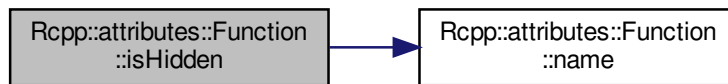
```
bool Rcpp::attributes::Function::isHidden ( ) const [inline]
```

Definition at line 262 of file attributes.cpp.

References name().

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions().

Here is the call graph for this function:



### 6.235.3.4 name()

```
const std::string& Rcpp::attributes::Function::name ( ) const [inline]
```

Definition at line 279 of file attributes.cpp.

References name\_.

Referenced by empty(), Rcpp::attributes::generateCpp(), isHidden(), renamedTo(), signature(), and Rcpp::attributes::↔ CppExportsGenerator::writeEnd().

### 6.235.3.5 operator"!=()"

```
bool Rcpp::attributes::Function::operator!= (
    const Function & other ) const [inline]
```

Definition at line 274 of file attributes.cpp.

### 6.235.3.6 operator==()

```
bool Rcpp::attributes::Function::operator== (
    const Function & other ) const [inline]
```

Definition at line 268 of file attributes.cpp.

References arguments\_, name\_, and type\_.

### 6.235.3.7 renamedTo()

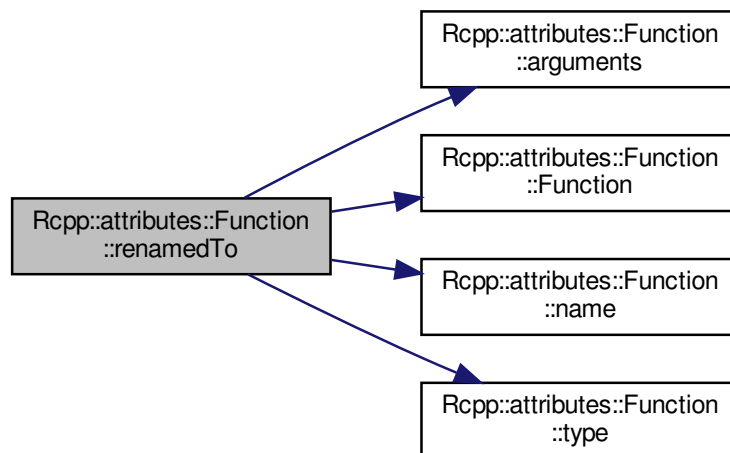
```
Function Rcpp::attributes::Function::renamedTo (
    const std::string & name ) const [inline]
```

Definition at line 255 of file attributes.cpp.

References arguments(), Function(), name(), and type().

Referenced by Rcpp::attributes::CppExportsGenerator::doWriteFunctions(), and Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions().

Here is the call graph for this function:



**6.235.3.8 signature()** [1/2]

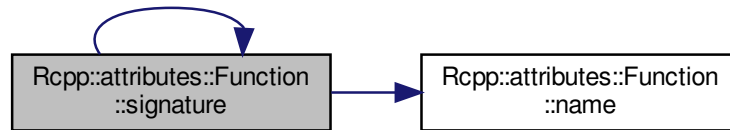
```
std::string Rcpp::attributes::Function::signature ( ) const [inline]
```

Definition at line 259 of file attributes.cpp.

References name(), and signature().

Referenced by signature(), and Rcpp::attributes::CppExportsGenerator::writeEnd().

Here is the call graph for this function:

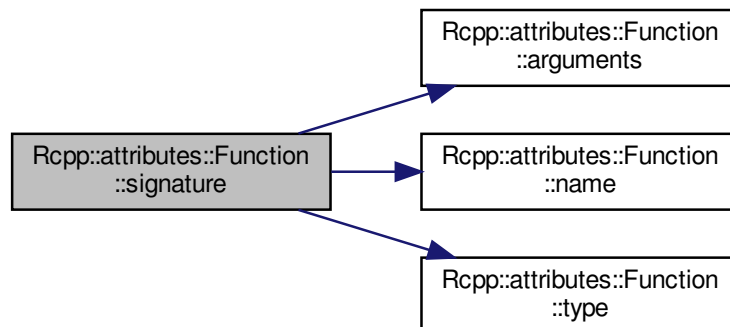
**6.235.3.9 signature()** [2/2]

```
std::string Rcpp::attributes::Function::signature (
    const std::string & name ) const
```

Definition at line 1042 of file attributes.cpp.

References arguments(), name(), and type().

Here is the call graph for this function:



### 6.235.3.10 type()

```
const Type& Rcpp::attributes::Function::type ( ) const [inline]
```

Definition at line 278 of file attributes.cpp.

References [type\\_](#).

Referenced by [renamedTo\(\)](#), and [signature\(\)](#).

## 6.235.4 Member Data Documentation

### 6.235.4.1 arguments\_

```
std::vector<Argument> Rcpp::attributes::Function::arguments_ [private]
```

Definition at line 285 of file attributes.cpp.

Referenced by [arguments\(\)](#), and [operator==\(\)](#).

### 6.235.4.2 name\_

```
std::string Rcpp::attributes::Function::name_ [private]
```

Definition at line 284 of file attributes.cpp.

Referenced by [name\(\)](#), and [operator==\(\)](#).

### 6.235.4.3 type\_

```
Type Rcpp::attributes::Function::type_ [private]
```

Definition at line 283 of file attributes.cpp.

Referenced by [operator==\(\)](#), and [type\(\)](#).

The documentation for this class was generated from the following file:

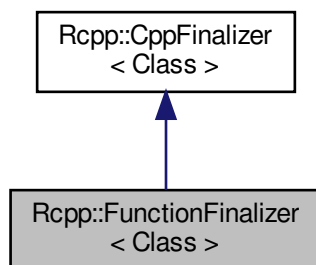
- [src/attributes.cpp](#)



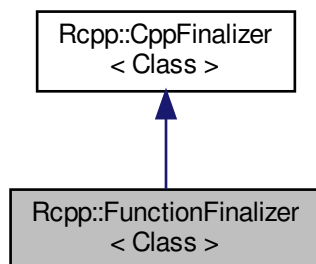
## 6.236 Rcpp::FunctionFinalizer< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::FunctionFinalizer< Class >:



Collaboration diagram for Rcpp::FunctionFinalizer< Class >:



### Public Types

- typedef void(\* [Pointer](#)) (Class \*)

### Public Member Functions

- [FunctionFinalizer](#) ([Pointer](#) p)
- virtual void [run](#) (Class \*object)

## Private Attributes

- [Pointer finalizer](#)

### 6.236.1 Detailed Description

```
template<typename Class >  
class Rcpp::FunctionFinalizer< Class >
```

Definition at line 307 of file Module.h.

### 6.236.2 Member Typedef Documentation

#### 6.236.2.1 Pointer

```
template<typename Class >  
typedef void(* Rcpp::FunctionFinalizer< Class >::Pointer) (Class *)
```

Definition at line 309 of file Module.h.

### 6.236.3 Constructor & Destructor Documentation

#### 6.236.3.1 FunctionFinalizer()

```
template<typename Class >  
Rcpp::FunctionFinalizer< Class >::FunctionFinalizer (  
    Pointer p ) [inline]
```

Definition at line 310 of file Module.h.

### 6.236.4 Member Function Documentation

### 6.236.4.1 run()

```
template<typename Class >
virtual void Rcpp::FunctionFinalizer< Class >::run (
    Class * object ) [inline], [virtual]
```

Reimplemented from [Rcpp::CppFinalizer< Class >](#).

Definition at line 312 of file Module.h.

References [Rcpp::FunctionFinalizer< Class >::finalizer](#).

## 6.236.5 Member Data Documentation

### 6.236.5.1 finalizer

```
template<typename Class >
Pointer Rcpp::FunctionFinalizer< Class >::finalizer [private]
```

Definition at line 317 of file Module.h.

Referenced by [Rcpp::FunctionFinalizer< Class >::run\(\)](#).

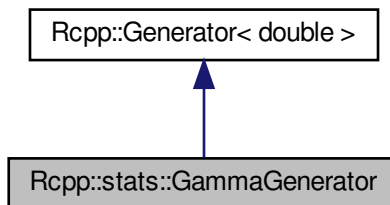
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Module.h](#)

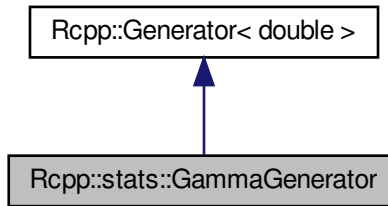
## 6.237 Rcpp::stats::GammaGenerator Class Reference

```
#include <rgamma.h>
```

Inheritance diagram for [Rcpp::stats::GammaGenerator](#):



Collaboration diagram for Rcpp::stats::GammaGenerator:



## Public Member Functions

- [GammaGenerator](#) (double *a\_*, double *scale\_*)
- double [operator\(\)](#) () const

## Private Attributes

- double [a](#)
- double [scale](#)

## Additional Inherited Members

### 6.237.1 Detailed Description

Definition at line 28 of file `rgamma.h`.

### 6.237.2 Constructor & Destructor Documentation

#### 6.237.2.1 GammaGenerator()

```
Rcpp::stats::GammaGenerator::GammaGenerator (  
    double a_,  
    double scale_ ) [inline]
```

Definition at line 30 of file `rgamma.h`.

### 6.237.3 Member Function Documentation

#### 6.237.3.1 operator()

```
double Rcpp::stats::GammaGenerator::operator() ( ) const [inline]
```

Definition at line 32 of file rgamma.h.

References `a`, and `scale`.

### 6.237.4 Member Data Documentation

#### 6.237.4.1 `a`

```
double Rcpp::stats::GammaGenerator::a [private]
```

Definition at line 34 of file rgamma.h.

Referenced by `operator()`.

#### 6.237.4.2 `scale`

```
double Rcpp::stats::GammaGenerator::scale [private]
```

Definition at line 34 of file rgamma.h.

Referenced by `operator()`.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rgamma.h](#)

## 6.238 Rcpp::Generator< T > Class Template Reference

```
#include <random.h>
```

## Public Types

- typedef T [r\\_generator](#)

### 6.238.1 Detailed Description

```
template<typename T>  
class Rcpp::Generator< T >
```

Definition at line 28 of file random.h.

### 6.238.2 Member Typedef Documentation

#### 6.238.2.1 r\_generator

```
template<typename T >  
typedef T Rcpp::Generator< T >::r\_generator
```

Definition at line 30 of file random.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/[random.h](#)

## 6.239 Rcpp::internal::generic\_element\_converter< RTYPE > Class Template Reference

```
#include <converter.h>
```

## Public Types

- typedef SEXP [target](#)

## Static Public Member Functions

- template<typename T >  
static SEXP [get](#) (const T &input)
- static SEXP [get](#) (const char \*input)
- static SEXP [get](#) (SEXP input)

## 6.239.1 Detailed Description

```
template<int RTYPE>
class Rcpp::internal::generic_element_converter< RTYPE >
```

Definition at line 74 of file converter.h.

## 6.239.2 Member Typedef Documentation

### 6.239.2.1 target

```
template<int RTYPE>
typedef SEXP Rcpp::internal::generic_element_converter< RTYPE >::target
```

Definition at line 76 of file converter.h.

## 6.239.3 Member Function Documentation

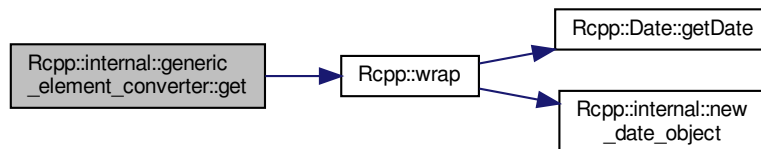
### 6.239.3.1 get() [1/3]

```
template<int RTYPE>
static SEXP Rcpp::internal::generic_element_converter< RTYPE >::get (
    const char * input ) [inline], [static]
```

Definition at line 83 of file converter.h.

References Rcpp::wrap().

Here is the call graph for this function:



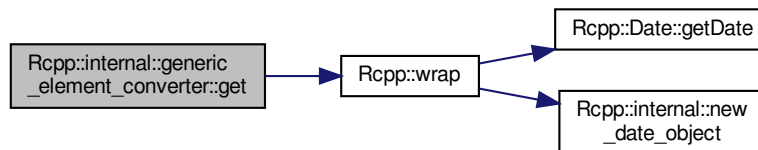
**6.239.3.2 get()** [2/3]

```
template<int RTYPE>
template<typename T >
static SEXP Rcpp::internal::generic_element_converter< RTYPE >::get (
    const T & input ) [inline], [static]
```

Definition at line 79 of file converter.h.

References Rcpp::wrap().

Here is the call graph for this function:

**6.239.3.3 get()** [3/3]

```
template<int RTYPE>
static SEXP Rcpp::internal::generic_element_converter< RTYPE >::get (
    SEXP input ) [inline], [static]
```

Definition at line 87 of file converter.h.

The documentation for this class was generated from the following file:

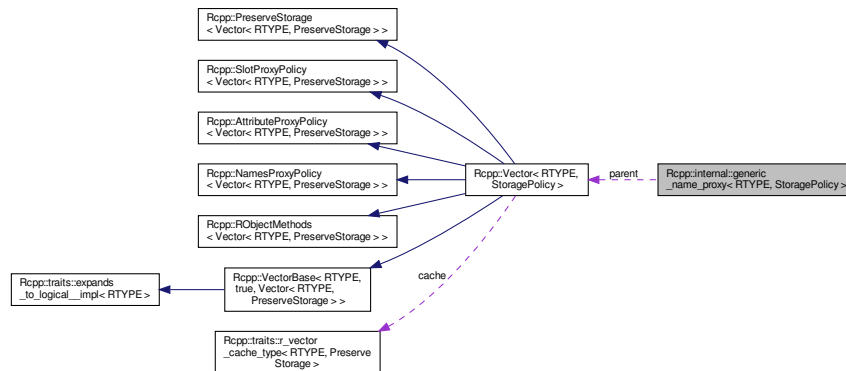
- [inst/include/Rcpp/vector/converter.h](#)



## 6.240 Rcpp::internal::generic\_name\_proxy< RTYPE, StoragePolicy > Class Template Reference

```
#include <proxy.h>
```

Collaboration diagram for Rcpp::internal::generic\_name\_proxy< RTYPE, StoragePolicy >:



### Public Types

- typedef `::Rcpp::Vector< RTYPE, StoragePolicy >` `VECTOR`

### Public Member Functions

- `generic_name_proxy` (`VECTOR &v`, `const std::string &name_`)
- `generic_name_proxy` (`const generic_name_proxy &other`)
- `~generic_name_proxy` ()
- `generic_name_proxy & operator=` (`SEXP rhs`)
- `generic_name_proxy & operator=` (`const generic_name_proxy &other`)
- `template<typename T >`  
`generic_name_proxy & operator=` (`const T &rhs`)
- `operator SEXP` () `const`
- `template<typename T >`  
`operator T` () `const`

### Private Member Functions

- `void set` (`SEXP rhs`)
- `SEXP get` () `const`

### Private Attributes

- `VECTOR & parent`
- `std::string name`

### 6.240.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>
class Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >
```

Definition at line 137 of file proxy.h.

### 6.240.2 Member Typedef Documentation

#### 6.240.2.1 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::VECTOR
```

Definition at line 139 of file proxy.h.

### 6.240.3 Constructor & Destructor Documentation

#### 6.240.3.1 generic\_name\_proxy() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::generic_name_proxy (
    VECTOR & v,
    const std::string & name_ ) [inline]
```

Definition at line 140 of file proxy.h.

#### 6.240.3.2 generic\_name\_proxy() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::generic_name_proxy (
    const generic_name_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 142 of file proxy.h.

### 6.240.3.3 ~generic\_name\_proxy()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::~~generic_name_proxy ( ) [inline]
```

Definition at line 144 of file proxy.h.

## 6.240.4 Member Function Documentation

### 6.240.4.1 get()

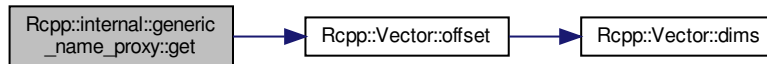
```
template<int RTYPE, template< class > class StoragePolicy>
SEXPR Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::get ( ) const [inline], [private]
```

Definition at line 190 of file proxy.h.

References `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::name`, `Rcpp::Vector< RTYPE, StoragePolicy >::offset()`, and `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::parent`.

Referenced by `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator SEXP()`, `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator T()`, and `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator=()`.

Here is the call graph for this function:



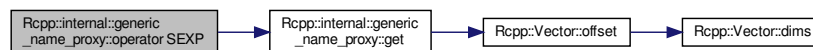
### 6.240.4.2 operator SEXP()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const [inline]
```

Definition at line 163 of file proxy.h.

References `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::get()`.

Here is the call graph for this function:



### 6.240.4.3 operator T()

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator T ( ) const [inline]
```

Definition at line 168 of file proxy.h.

References `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::get()`, and `RCPP_DEBUG_1`.

Here is the call graph for this function:



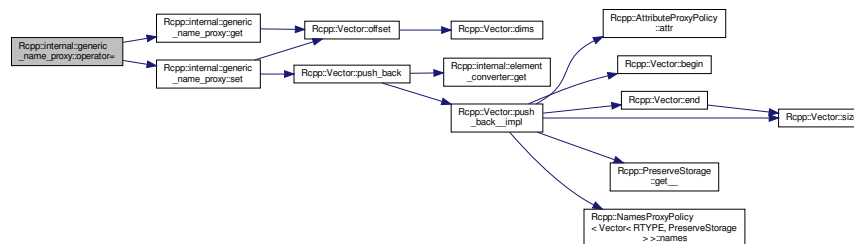
### 6.240.4.4 operator=() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
generic_name_proxy& Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator= (
    const generic_name_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 150 of file proxy.h.

References `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::get()`, and `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::set()`.

Here is the call graph for this function:



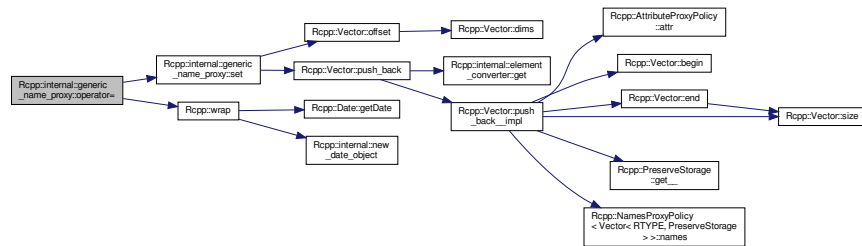
## 6.240.4.5 operator=() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
generic_name_proxy& Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator= (
    const T & rhs ) [inline]
```

Definition at line 156 of file proxy.h.

References Rcpp::internal::generic\_name\_proxy< RTYPE, StoragePolicy >::set(), and Rcpp::wrap().

Here is the call graph for this function:



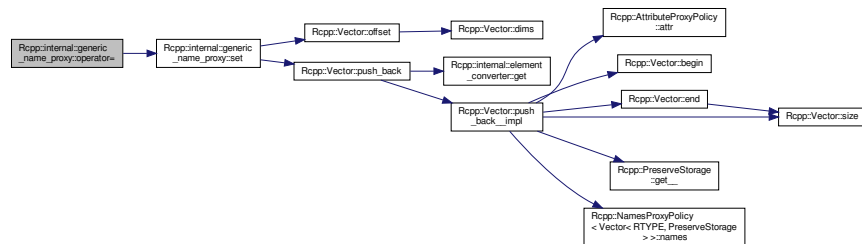
## 6.240.4.6 operator=() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
generic_name_proxy& Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator= (
    SEXP rhs ) [inline]
```

Definition at line 146 of file proxy.h.

References Rcpp::internal::generic\_name\_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



### 6.240.4.7 set()

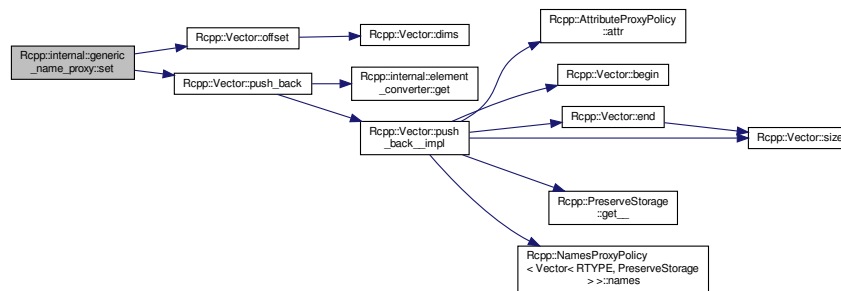
```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::set (
    SEXP rhs ) [inline], [private]
```

Definition at line 181 of file proxy.h.

References `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::name`, `Rcpp::Vector< RTYPE, StoragePolicy >::offset()`, `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::parent`, and `Rcpp::Vector< RTYPE, StoragePolicy >::push_back()`.

Referenced by `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::operator=()`.

Here is the call graph for this function:



## 6.240.5 Member Data Documentation

### 6.240.5.1 name

```
template<int RTYPE, template< class > class StoragePolicy>
std::string Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::name [private]
```

Definition at line 180 of file proxy.h.

Referenced by `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::get()`, and `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::set()`.

### 6.240.5.2 parent

```
template<int RTYPE, template< class > class StoragePolicy>
VECTOR& Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::parent [private]
```

Definition at line 179 of file proxy.h.

Referenced by Rcpp::internal::generic\_name\_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::generic\_name\_proxy< RTYPE, StoragePolicy >::set().

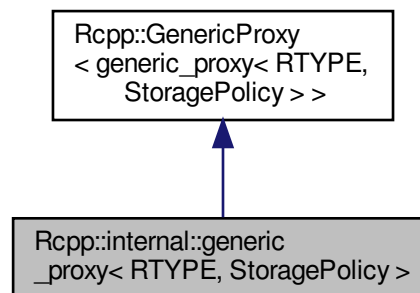
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00\_forward\_proxy.h
- inst/include/Rcpp/vector/proxy.h

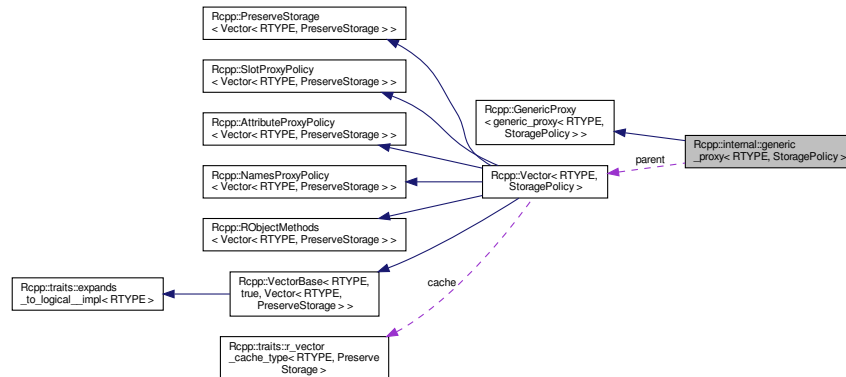
## 6.241 Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy > Class Template Reference

```
#include <generic_proxy.h>
```

Inheritance diagram for Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >:



Collaboration diagram for `Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >`:



## Public Types

- typedef `::Rcpp::Vector< RTYPE, StoragePolicy >` `VECTOR`

## Public Member Functions

- `generic_proxy ()`
- `generic_proxy (const generic_proxy &other)`
- `generic_proxy (VECTOR &v, R_xlen_t i)`
- `generic_proxy & operator= (SEXP rhs)`
- `generic_proxy & operator= (const generic_proxy &rhs)`
- `template<template< class > class StoragePolicy2>`  
`generic_proxy & operator= (const generic_proxy< RTYPE, StoragePolicy2 > &rhs)`
- `template<typename T >`  
`generic_proxy & operator= (const T &rhs)`
- `operator SEXP () const`
- `template<typename U >`  
`operator U () const`
- `operator bool () const`
- `operator int () const`
- `void swap (generic_proxy &other)`
- `void move (R_xlen_t n)`
- `void import (const generic_proxy &other)`
- `SEXP get () const`

## Public Attributes

- `VECTOR` \* `parent`
- `R_xlen_t` `index`



## Private Member Functions

- void [set](#) (SEXP x)

### 6.241.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>
class Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >
```

Definition at line 27 of file generic\_proxy.h.

### 6.241.2 Member Typedef Documentation

#### 6.241.2.1 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::generic_proxy< RTYPE, StoragePolicy
>::VECTOR
```

Definition at line 29 of file generic\_proxy.h.

### 6.241.3 Constructor & Destructor Documentation

#### 6.241.3.1 generic\_proxy() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::generic_proxy ( ) [inline]
```

Definition at line 31 of file generic\_proxy.h.

#### 6.241.3.2 generic\_proxy() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::generic_proxy (
    const generic_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 33 of file generic\_proxy.h.

### 6.241.3.3 generic\_proxy() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::generic_proxy (
    VECTOR & v,
    R_xlen_t i ) [inline]
```

Definition at line 37 of file generic\_proxy.h.

## 6.241.4 Member Function Documentation

### 6.241.4.1 get()

```
template<int RTYPE, template< class > class StoragePolicy>
SEXP Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get ( ) const [inline]
```

Definition at line 90 of file generic\_proxy.h.

References [Rcpp::internal::generic\\_proxy< RTYPE, StoragePolicy >::index](#), and [Rcpp::internal::generic\\_proxy< RTYPE, StoragePolicy >::parent](#).

Referenced by [Rcpp::internal::generic\\_proxy< RTYPE, StoragePolicy >::operator bool\(\)](#), [Rcpp::internal::generic\\_proxy< RTYPE, StoragePolicy >::operator int\(\)](#), [Rcpp::internal::generic\\_proxy< RTYPE, StoragePolicy >::operator SEXP\(\)](#), [Rcpp::internal::generic\\_proxy< RTYPE, StoragePolicy >::operator U\(\)](#), [Rcpp::internal::generic\\_proxy< RTYPE, StoragePolicy >::operator=\(\)](#), and [Rcpp::internal::generic\\_proxy< RTYPE, StoragePolicy >::swap\(\)](#).

### 6.241.4.2 import()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::import (
    const generic_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 85 of file generic\_proxy.h.

References [Rcpp::internal::generic\\_proxy< RTYPE, StoragePolicy >::index](#), and [Rcpp::internal::generic\\_proxy< RTYPE, StoragePolicy >::parent](#).

### 6.241.4.3 move()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::move (
    R_xlen_t n ) [inline]
```

Definition at line 83 of file generic\_proxy.h.

References Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::index.

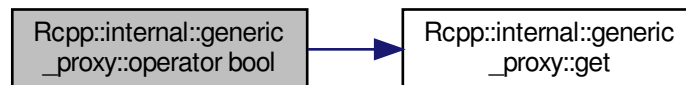
### 6.241.4.4 operator bool()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator bool ( ) const [inline]
```

Definition at line 72 of file generic\_proxy.h.

References Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



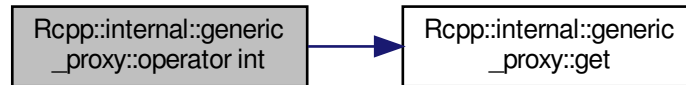
### 6.241.4.5 operator int()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator int ( ) const [inline]
```

Definition at line 73 of file generic\_proxy.h.

References Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



#### 6.241.4.6 operator SEXP()

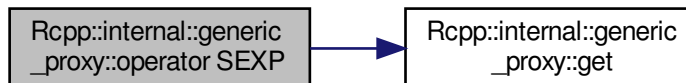
```

template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const [inline]
  
```

Definition at line 63 of file generic\_proxy.h.

References Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



#### 6.241.4.7 operator U()

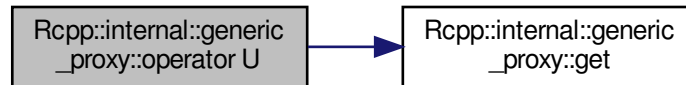
```

template<int RTYPE, template< class > class StoragePolicy>
template<typename U >
Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator U ( ) const [inline]
  
```

Definition at line 67 of file generic\_proxy.h.

References Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



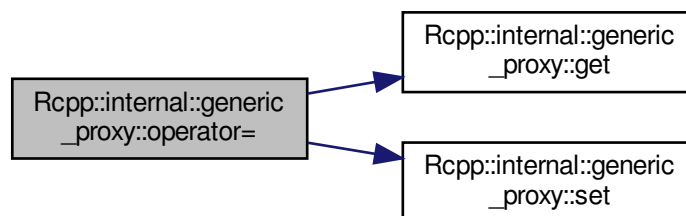
#### 6.241.4.8 operator=() [1/4]

```
template<int RTYPE, template< class > class StoragePolicy>  
generic_proxy& Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator= (  
    const generic_proxy< RTYPE, StoragePolicy > & rhs ) [inline]
```

Definition at line 46 of file `generic_proxy.h`.

References `Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get()`, and `Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::set()`.

Here is the call graph for this function:



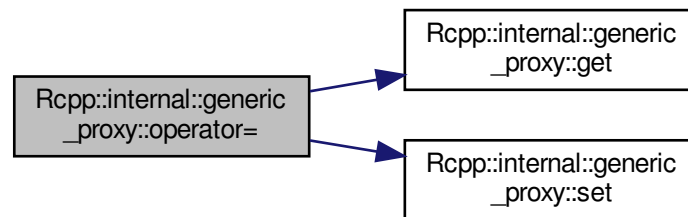
**6.241.4.9 operator=()** [2/4]

```
template<int RTYPE, template< class > class StoragePolicy>
template<template< class > class StoragePolicy2>
generic_proxy& Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator= (
    const generic_proxy< RTYPE, StoragePolicy2 > & rhs ) [inline]
```

Definition at line 52 of file generic\_proxy.h.

References Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:

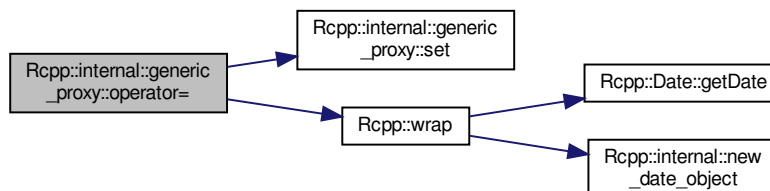
**6.241.4.10 operator=()** [3/4]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
generic_proxy& Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator= (
    const T & rhs ) [inline]
```

Definition at line 58 of file generic\_proxy.h.

References Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::set(), and Rcpp::wrap().

Here is the call graph for this function:



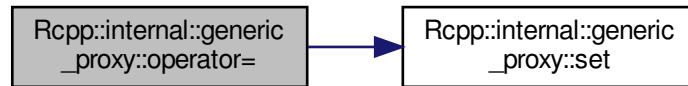
**6.241.4.11 operator=()** [4/4]

```
template<int RTYPE, template< class > class StoragePolicy>
generic_proxy& Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::operator= (
    SEXP rhs ) [inline]
```

Definition at line 41 of file generic\_proxy.h.

References Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:

**6.241.4.12 set()**

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::set (
    SEXP x ) [inline], [private]
```

Definition at line 95 of file generic\_proxy.h.

References Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::operator=(), and Rcpp::internal::generic\_proxy< RTYPE, StoragePolicy >::swap().

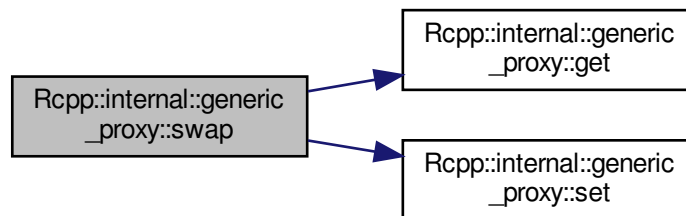
### 6.241.4.13 swap()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::swap (
    generic_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 75 of file generic\_proxy.h.

References `Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get()`, and `Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::set()`.

Here is the call graph for this function:



## 6.241.5 Member Data Documentation

### 6.241.5.1 index

```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::index
```

Definition at line 82 of file generic\_proxy.h.

Referenced by `Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get()`, `Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::import()`, `Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::move()`, and `Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::set()`.



### 6.241.5.2 parent

```
template<int RTYPE, template< class > class StoragePolicy>
VECTOR* Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::parent
```

Definition at line 81 of file generic\_proxy.h.

Referenced by `Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::get()`, `Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::import()`, and `Rcpp::internal::generic_proxy< RTYPE, StoragePolicy >::set()`.

The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00\_forward\_proxy.h
- inst/include/Rcpp/vector/generic\_proxy.h

## 6.242 Rcpp::GenericProxy< Proxy > Struct Template Reference

```
#include <GenericProxy.h>
```

### Public Member Functions

- SEXP `get` () const

### 6.242.1 Detailed Description

```
template<typename Proxy>
struct Rcpp::GenericProxy< Proxy >
```

Definition at line 24 of file GenericProxy.h.

### 6.242.2 Member Function Documentation

#### 6.242.2.1 get()

```
template<typename Proxy >
SEXP Rcpp::GenericProxy< Proxy >::get ( ) const [inline]
```

Definition at line 25 of file GenericProxy.h.

Referenced by `Rcpp::Vector< RTYPE, StoragePolicy >::Vector()`.

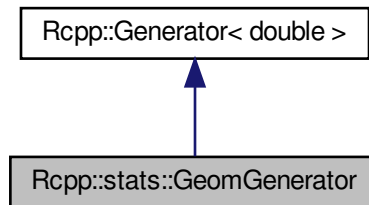
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/proxy/GenericProxy.h

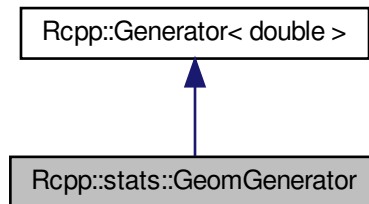
## 6.243 Rcpp::stats::GeomGenerator Class Reference

```
#include <rgeom.h>
```

Inheritance diagram for Rcpp::stats::GeomGenerator:



Collaboration diagram for Rcpp::stats::GeomGenerator:



### Public Member Functions

- [GeomGenerator](#) (double p)
- double [operator\(\)](#) () const

### Private Attributes

- double [lambda](#)

## Additional Inherited Members

### 6.243.1 Detailed Description

Definition at line 28 of file rgeom.h.

### 6.243.2 Constructor & Destructor Documentation

#### 6.243.2.1 GeomGenerator()

```
Rcpp::stats::GeomGenerator::GeomGenerator (  
    double p ) [inline]
```

Definition at line 31 of file rgeom.h.

### 6.243.3 Member Function Documentation

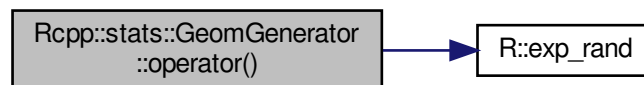
#### 6.243.3.1 operator()()

```
double Rcpp::stats::GeomGenerator::operator() ( ) const [inline]
```

Definition at line 33 of file rgeom.h.

References [R::exp\\_rand\(\)](#), and [lambda](#).

Here is the call graph for this function:



## 6.243.4 Member Data Documentation

### 6.243.4.1 lambda

```
double Rcpp::stats::GeomGenerator::lambda [private]
```

Definition at line 38 of file rgeom.h.

Referenced by operator().

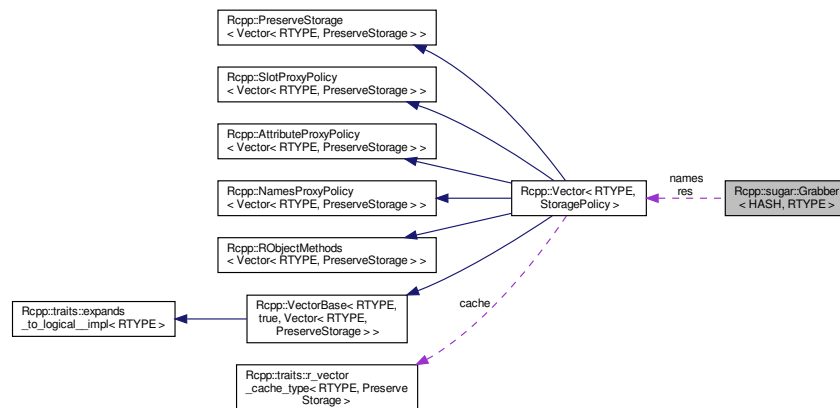
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rgeom.h](#)

## 6.244 Rcpp::sugar::Grabber< HASH, RTYPE > Class Template Reference

```
#include <table.h>
```

Collaboration diagram for Rcpp::sugar::Grabber< HASH, RTYPE >:



## Public Member Functions

- [Grabber](#) ([IntegerVector](#) &res\_, [CharacterVector](#) &names\_)
- `template<typename T >`  
void [operator\(\)](#) (T pair)

## Private Attributes

- [IntegerVector](#) & *res*
- [CharacterVector](#) & *names*
- [R\\_xlen\\_t](#) *index*

### 6.244.1 Detailed Description

```
template<typename HASH, int RTYPE>
class Rcpp::sugar::Grabber< HASH, RTYPE >
```

Definition at line 42 of file table.h.

### 6.244.2 Constructor & Destructor Documentation

#### 6.244.2.1 Grabber()

```
template<typename HASH , int RTYPE>
Rcpp::sugar::Grabber< HASH, RTYPE >::Grabber (
    IntegerVector & res_,
    CharacterVector & names_ ) [inline]
```

Definition at line 44 of file table.h.

### 6.244.3 Member Function Documentation

#### 6.244.3.1 operator>()()

```
template<typename HASH , int RTYPE>
template<typename T >
void Rcpp::sugar::Grabber< HASH, RTYPE >::operator() (
    T pair ) [inline]
```

Definition at line 47 of file table.h.

References [Rcpp::sugar::Grabber< HASH, RTYPE >::index](#), [Rcpp::sugar::Grabber< HASH, RTYPE >::names](#), and [Rcpp::sugar::Grabber< HASH, RTYPE >::res](#).

## 6.244.4 Member Data Documentation

### 6.244.4.1 index

```
template<typename HASH , int RTYPE>  
R_xlen_t Rcpp::sugar::Grabber< HASH, RTYPE >::index [private]
```

Definition at line 55 of file table.h.

Referenced by Rcpp::sugar::Grabber< HASH, RTYPE >::operator()().

### 6.244.4.2 names

```
template<typename HASH , int RTYPE>  
CharacterVector& Rcpp::sugar::Grabber< HASH, RTYPE >::names [private]
```

Definition at line 54 of file table.h.

Referenced by Rcpp::sugar::Grabber< HASH, RTYPE >::operator()().

### 6.244.4.3 res

```
template<typename HASH , int RTYPE>  
IntegerVector& Rcpp::sugar::Grabber< HASH, RTYPE >::res [private]
```

Definition at line 53 of file table.h.

Referenced by Rcpp::sugar::Grabber< HASH, RTYPE >::operator()().

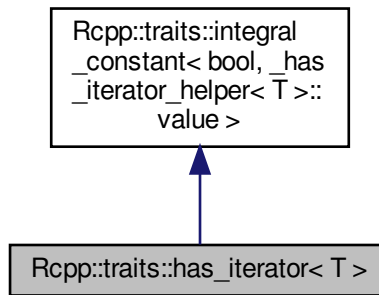
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[table.h](#)

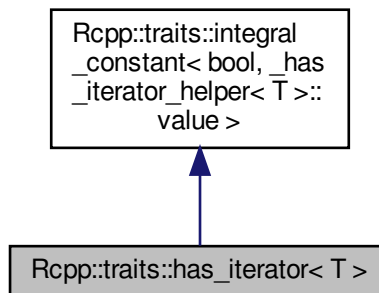
## 6.245 Rcpp::traits::has\_iterator< T > Struct Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::has\_iterator< T >:



Collaboration diagram for Rcpp::traits::has\_iterator< T >:



### Additional Inherited Members

#### 6.245.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::has_iterator< T >
```

uses the SFINAE idiom to check if a class has an nested iterator typedef. For example :

[has\\_iterator< std::vector<int> >::value](#) is true [has\\_iterator< Rcpp::Symbol >::value](#) is false

Definition at line 103 of file `has_iterator.h`.

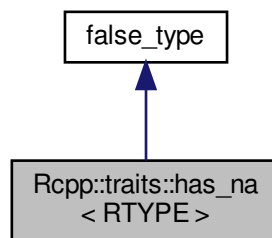
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/has_iterator.h`

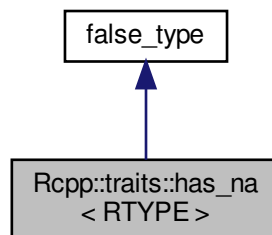
## 6.246 Rcpp::traits::has\_na< RTYPE > Struct Template Reference

```
#include <has_na.h>
```

Inheritance diagram for `Rcpp::traits::has_na< RTYPE >`:



Collaboration diagram for `Rcpp::traits::has_na< RTYPE >`:





## Additional Inherited Members

### 6.246.1 Detailed Description

```
template<int RTYPE>  
struct Rcpp::traits::has_na< RTYPE >
```

Identifies if a given SEXP type has the concept of missing values

This is false by default and specialized for all types that do have the concept

Definition at line 35 of file `has_na.h`.

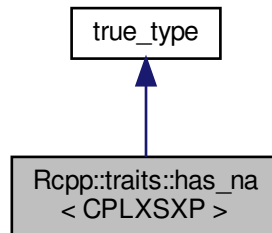
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/has_na.h`

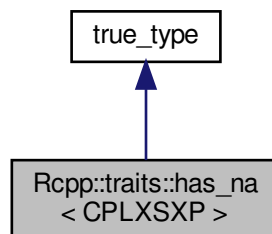
## 6.247 Rcpp::traits::has\_na< CPLXSXP > Struct Reference

```
#include <has_na.h>
```

Inheritance diagram for `Rcpp::traits::has_na< CPLXSXP >`:



Collaboration diagram for `Rcpp::traits::has_na< CPLXSXP >`:



## Additional Inherited Members

### 6.247.1 Detailed Description

complex vectors support missing values

Definition at line 50 of file `has_na.h`.

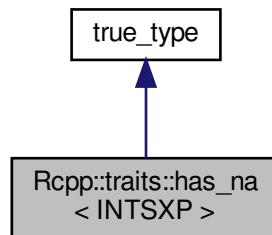
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/has_na.h`

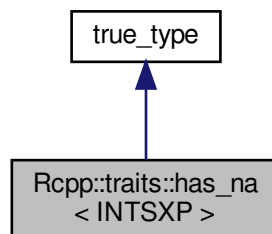
### 6.248 Rcpp::traits::has\_na< INTSXP > Struct Reference

```
#include <has_na.h>
```

Inheritance diagram for Rcpp::traits::has\_na< INTSXP >:



Collaboration diagram for Rcpp::traits::has\_na< INTSXP >:



## Additional Inherited Members

### 6.248.1 Detailed Description

integer vectors support missing values

Definition at line 40 of file `has_na.h`.

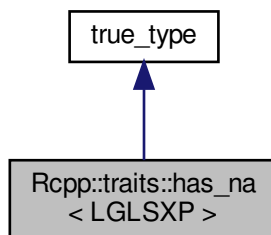
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/has_na.h`

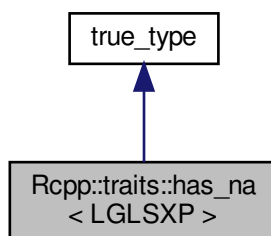
## 6.249 Rcpp::traits::has\_na< LGLSXP > Struct Reference

```
#include <has_na.h>
```

Inheritance diagram for `Rcpp::traits::has_na< LGLSXP >`:



Collaboration diagram for `Rcpp::traits::has_na< LGLSXP >`:



## Additional Inherited Members

### 6.249.1 Detailed Description

logical vectors support missing values

Definition at line 60 of file `has_na.h`.

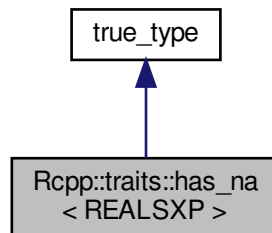
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/has_na.h`

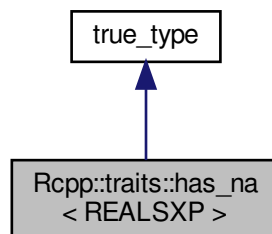
### 6.250 Rcpp::traits::has\_na< REALSXP > Struct Reference

```
#include <has_na.h>
```

Inheritance diagram for Rcpp::traits::has\_na< REALSXP >:



Collaboration diagram for Rcpp::traits::has\_na< REALSXP >:



## Additional Inherited Members

### 6.250.1 Detailed Description

numeric vectors support missing values

Definition at line 45 of file `has_na.h`.

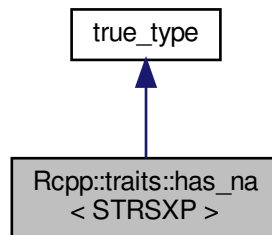
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/has_na.h`

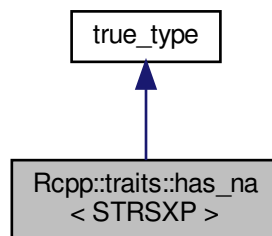
## 6.251 Rcpp::traits::has\_na< STRSXP > Struct Reference

```
#include <has_na.h>
```

Inheritance diagram for `Rcpp::traits::has_na< STRSXP >`:



Collaboration diagram for `Rcpp::traits::has_na< STRSXP >`:



## Additional Inherited Members

### 6.251.1 Detailed Description

character vector support missing values

Definition at line 55 of file `has_na.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/has_na.h`

## 6.252 `Rcpp::sugar::cbind_impl::detail::has_stored_type< T >` Class Template Reference

```
#include <cbind.h>
```

### Classes

- struct `no`

### Static Public Attributes

- static const bool `value` = `sizeof(test<T>(0)) == sizeof(yes)`

### Private Types

- typedef char `yes`

### Static Private Member Functions

- `template<typename C >`  
static `yes test` (`typename C::stored_type *`)
- `template<typename C >`  
static `no test` (...)

### 6.252.1 Detailed Description

```
template<typename T>  
class Rcpp::sugar::cbind_impl::detail::has_stored_type< T >
```

Definition at line 373 of file `cbind.h`.

## 6.252.2 Member Typedef Documentation

### 6.252.2.1 yes

```
template<typename T >
typedef char Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::yes [private]
```

Definition at line 375 of file cbind.h.

## 6.252.3 Member Function Documentation

### 6.252.3.1 test() [1/2]

```
template<typename T >
template<typename C >
static no Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::test (
    ... ) [static], [private]
```

### 6.252.3.2 test() [2/2]

```
template<typename T >
template<typename C >
static yes Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::test (
    typename C::stored_type * ) [static], [private]
```

## 6.252.4 Member Data Documentation

### 6.252.4.1 value

```
template<typename T >
const bool Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::value = sizeof(test<T>(0)) ==
sizeof(yes) [static]
```

Definition at line 387 of file cbind.h.

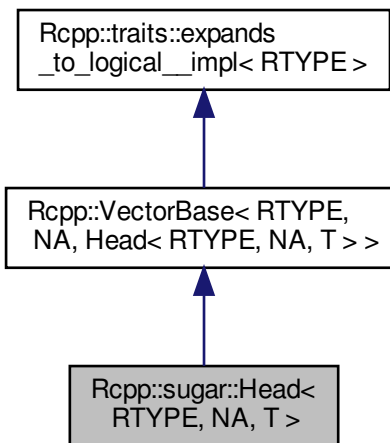
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

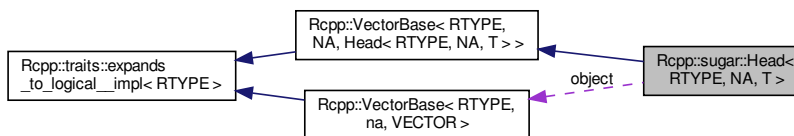
## 6.253 Rcpp::sugar::Head< RTYPE, NA, T > Class Template Reference

```
#include <head.h>
```

Inheritance diagram for Rcpp::sugar::Head< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Head< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`

### Public Member Functions

- `Head` (const `VEC_TYPE` &object\_, `R_xlen_t` n\_)
- `STORAGE` `operator[]` (`R_xlen_t` i) const
- `R_xlen_t` `size` () const



## Private Attributes

- const [VEC\\_TYPE](#) & *object*
- int *n*

### 6.253.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Head< RTYPE, NA, T >
```

Definition at line 29 of file head.h.

### 6.253.2 Member Typedef Documentation

#### 6.253.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Head< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file head.h.

#### 6.253.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Head< RTYPE, NA, T >::VEC\_TYPE
```

Definition at line 31 of file head.h.

### 6.253.3 Constructor & Destructor Documentation

#### 6.253.3.1 Head()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Head< RTYPE, NA, T >::Head (
    const VEC\_TYPE & object_,
    R\_xlen\_t n_ ) [inline]
```

Definition at line 34 of file head.h.

References [Rcpp::sugar::Head< RTYPE, NA, T >::n](#).

## 6.253.4 Member Function Documentation

### 6.253.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >  
STORAGE Rcpp::sugar::Head< RTYPE, NA, T >::operator[] (  
    R_xlen_t i ) const [inline]
```

Definition at line 40 of file head.h.

### 6.253.4.2 size()

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Head< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 43 of file head.h.

References `Rcpp::sugar::Head< RTYPE, NA, T >::n`.

## 6.253.5 Member Data Documentation

### 6.253.5.1 n

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::sugar::Head< RTYPE, NA, T >::n [private]
```

Definition at line 47 of file head.h.

Referenced by `Rcpp::sugar::Head< RTYPE, NA, T >::Head()`, and `Rcpp::sugar::Head< RTYPE, NA, T >::size()`.

### 6.253.5.2 object

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::sugar::Head< RTYPE, NA, T >::object [private]
```

Definition at line 46 of file head.h.

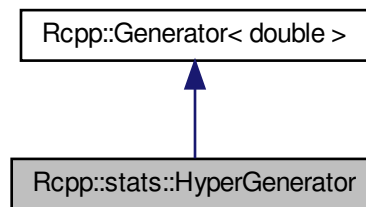
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/head.h`

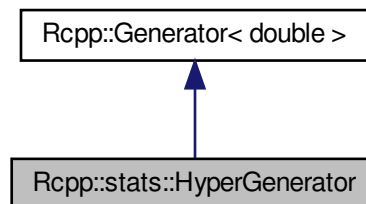
## 6.254 Rcpp::stats::HyperGenerator Class Reference

```
#include <rhyper.h>
```

Inheritance diagram for Rcpp::stats::HyperGenerator:



Collaboration diagram for Rcpp::stats::HyperGenerator:



### Public Member Functions

- [HyperGenerator](#) (double nn1\_, double nn2\_, double kk\_)
- double [operator\(\)](#) () const

### Private Attributes

- double [nn1](#)
- double [nn2](#)
- double [kk](#)

## Additional Inherited Members

### 6.254.1 Detailed Description

Definition at line 28 of file rhyper.h.

### 6.254.2 Constructor & Destructor Documentation

#### 6.254.2.1 HyperGenerator()

```
Rcpp::stats::HyperGenerator::HyperGenerator (
    double nn1_,
    double nn2_,
    double kk_ ) [inline]
```

Definition at line 30 of file rhyper.h.

### 6.254.3 Member Function Documentation

#### 6.254.3.1 operator()()

```
double Rcpp::stats::HyperGenerator::operator() ( ) const [inline]
```

Definition at line 32 of file rhyper.h.

References `kk`, `nn1`, and `nn2`.

### 6.254.4 Member Data Documentation

#### 6.254.4.1 `kk`

```
double Rcpp::stats::HyperGenerator::kk [private]
```

Definition at line 34 of file rhyper.h.

Referenced by `operator()()`.

### 6.254.4.2 nn1

```
double Rcpp::stats::HyperGenerator::nn1 [private]
```

Definition at line 34 of file rhyper.h.

Referenced by operator().

### 6.254.4.3 nn2

```
double Rcpp::stats::HyperGenerator::nn2 [private]
```

Definition at line 34 of file rhyper.h.

Referenced by operator().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rhyper.h](#)

## 6.255 Rcpp::traits::identity< T > Struct Template Reference

```
#include <traits.h>
```

### Public Types

- typedef T [type](#)

### 6.255.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::identity< T >
```

Definition at line 30 of file traits.h.

### 6.255.2 Member Typedef Documentation

### 6.255.2.1 type

```
template<typename T >
typedef T Rcpp::traits::identity< T >::type
```

Definition at line 30 of file traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/traits.h

## 6.256 Rcpp::traits::if\_< COND, LHS, RHS > Struct Template Reference

```
#include <if_.h>
```

### Public Types

- typedef LHS [type](#)

### 6.256.1 Detailed Description

```
template<bool COND, typename LHS, typename RHS>
struct Rcpp::traits::if_< COND, LHS, RHS >
```

Definition at line 29 of file if\_.h.

### 6.256.2 Member Typedef Documentation

#### 6.256.2.1 type

```
template<bool COND, typename LHS , typename RHS >
typedef LHS Rcpp::traits::if_< COND, LHS, RHS >::type
```

Definition at line 30 of file if\_.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/if\_.h

## 6.257 Rcpp::traits::if\_< false, LHS, RHS > Struct Template Reference

```
#include <if_.h>
```

### Public Types

- typedef RHS [type](#)

### 6.257.1 Detailed Description

```
template<typename LHS, typename RHS>  
struct Rcpp::traits::if_< false, LHS, RHS >
```

Definition at line 34 of file if\_.h.

### 6.257.2 Member Typedef Documentation

#### 6.257.2.1 type

```
template<typename LHS , typename RHS >  
typedef RHS Rcpp::traits::if\_< false, LHS, RHS >::type
```

Definition at line 35 of file if\_.h.

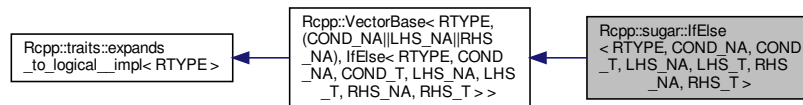
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/if\_.h

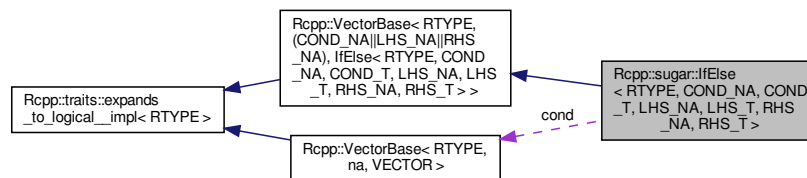
## 6.258 Rcpp::sugar::IfElse< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::IfElse< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase< LGLSXP, COND\\_NA, COND\\_T >](#) [COND\\_TYPE](#)
- typedef [Rcpp::VectorBase< RTYPE,LHS\\_NA,LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase< RTYPE,RHS\\_NA,RHS\\_T >](#) [RHS\\_TYPE](#)
- typedef [traits::storage\\_type< RTYPE >::type](#) [STORAGE](#)

### Public Member Functions

- [IfElse](#) (const [COND\\_TYPE](#) &cond\_, const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [COND\\_TYPE](#) & [cond](#)
- const [LHS\\_T](#) & [lhs](#)
- const [RHS\\_T](#) & [rhs](#)



## 6.258.1 Detailed Description

```
template<int RTYPE, bool COND_NA, typename COND_T, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 34 of file ifelse.h.

## 6.258.2 Member Typedef Documentation

### 6.258.2.1 COND\_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,COND_NA,COND_T> Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T,
LHS_NA, LHS_T, RHS_NA, RHS_T >::COND_TYPE
```

Definition at line 40 of file ifelse.h.

### 6.258.2.2 LHS\_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
typedef Rcpp::VectorBase<RTYPE ,LHS_NA ,LHS_T> Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS↔
_NA, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 41 of file ifelse.h.

### 6.258.2.3 RHS\_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
typedef Rcpp::VectorBase<RTYPE ,RHS_NA ,RHS_T> Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS↔
_NA, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 42 of file ifelse.h.

### 6.258.2.4 STORAGE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA,
LHS_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 43 of file ifelse.h.

## 6.258.3 Constructor & Destructor Documentation

### 6.258.3.1 IfElse()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::IfElse (
    const COND_TYPE & cond_,
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 48 of file ifelse.h.

References DEMANGLE, and RCPP\_DEBUG.

## 6.258.4 Member Function Documentation

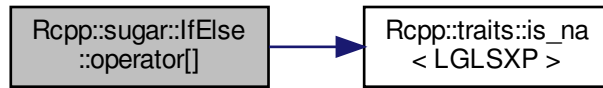
### 6.258.4.1 operator[]()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
STORAGE Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 55 of file ifelse.h.

References Rcpp::sugar::IfElse< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::cond, Rcpp::traits::is\_na< LGLSXP >(), Rcpp::sugar::IfElse< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::IfElse< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

Here is the call graph for this function:



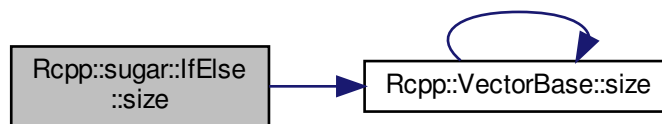
### 6.258.4.2 size()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
R_xlen_t Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( )
const [inline]
```

Definition at line 62 of file ifelse.h.

References Rcpp::sugar::IfElse< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::cond, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



### 6.258.5 Member Data Documentation

### 6.258.5.1 cond

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
const COND_TYPE& Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >←
::cond [private]
```

Definition at line 65 of file ifelse.h.

Referenced by `Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >←::operator[]()`, `Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >←::operator[]()`, `Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >←::size()`, and `Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >←::size()`.

### 6.258.5.2 lhs

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
const LHS_T& Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >←::lhs
[private]
```

Definition at line 66 of file ifelse.h.

Referenced by `Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >←::operator[]()`, and `Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >←::operator[]()`.

### 6.258.5.3 rhs

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA,
typename RHS_T >
const RHS_T& Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >←::rhs
[private]
```

Definition at line 67 of file ifelse.h.

Referenced by `Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >←::operator[]()`, and `Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >←::operator[]()`.

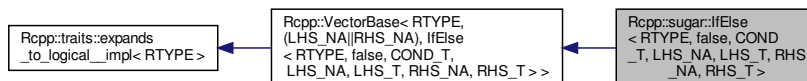
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/ifelse.h](#)

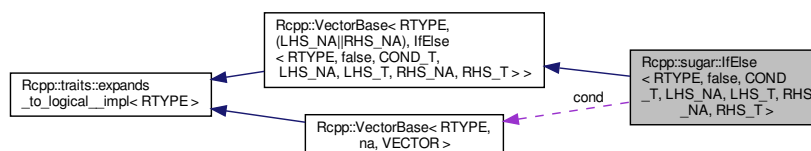
## 6.259 Rcpp::sugar::IfElse< RTYPE, false, COND\_T, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse< RTYPE, false, COND\_T, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::IfElse< RTYPE, false, COND\_T, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase< LGLSXP, false, COND\\_T >](#) [COND\\_TYPE](#)
- typedef [Rcpp::VectorBase< RTYPE,LHS\\_NA,LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase< RTYPE,RHS\\_NA,RHS\\_T >](#) [RHS\\_TYPE](#)
- typedef [traits::storage\\_type< RTYPE >::type](#) [STORAGE](#)
- typedef [Rcpp::traits::Extractor< RTYPE,LHS\\_NA,LHS\\_T >::type](#) [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor< RTYPE,RHS\\_NA,RHS\\_T >::type](#) [RHS\\_EXT](#)

### Public Member Functions

- [IfElse](#) (const [COND\\_TYPE](#) &cond\_, const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [COND\\_TYPE](#) & [cond](#)
- const [LHS\\_EXT](#) & [lhs](#)
- const [RHS\\_EXT](#) & [rhs](#)

### 6.259.1 Detailed Description

```
template<int RTYPE, typename COND_T, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>  
class Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_T, RHS_NA, RHS_T >
```

Definition at line 77 of file ifelse.h.

### 6.259.2 Member Typedef Documentation

#### 6.259.2.1 COND\_TYPE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::VectorBase<LGLSXP, false, COND_T> Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA,  
LHS_T, RHS_NA, RHS_T >::COND_TYPE
```

Definition at line 83 of file ifelse.h.

#### 6.259.2.2 LHS\_EXT

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::traits::Extractor<RTYPE ,LHS_NA ,LHS_T>::type Rcpp::sugar::IfElse< RTYPE, false,  
COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 88 of file ifelse.h.

#### 6.259.2.3 LHS\_TYPE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::VectorBase<RTYPE ,LHS_NA ,LHS_T> Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA,  
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 84 of file ifelse.h.

### 6.259.2.4 RHS\_EXT

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE ,RHS_NA ,RHS_T>::type Rcpp::sugar::IfElse< RTYPE, false,
COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 89 of file ifelse.h.

### 6.259.2.5 RHS\_TYPE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE ,RHS_NA ,RHS_T> Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 85 of file ifelse.h.

### 6.259.2.6 STORAGE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA,
LHS_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 86 of file ifelse.h.

## 6.259.3 Constructor & Destructor Documentation

### 6.259.3.1 IfElse()

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::IfElse (
    const COND_TYPE & cond_,
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 91 of file ifelse.h.

### 6.259.4 Member Function Documentation

### 6.259.4.1 operator[]()

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 96 of file ifelse.h.

References `Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::cond`, `Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

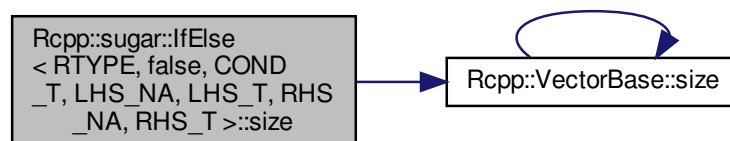
### 6.259.4.2 size()

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( )
const [inline]
```

Definition at line 101 of file ifelse.h.

References `Rcpp::sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::cond`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`.

Here is the call graph for this function:



## 6.259.5 Member Data Documentation

### 6.259.5.1 cond

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const COND_TYPE& Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::cond
[private]
```

Definition at line 105 of file ifelse.h.



6.259.5.2 lhs

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 106 of file ifelse.h.

6.259.5.3 rhs

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::IfElse< RTYPE, false, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 107 of file ifelse.h.

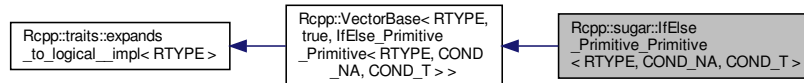
The documentation for this class was generated from the following file:

- <inst/include/Rcpp/sugar/functions/ifelse.h>

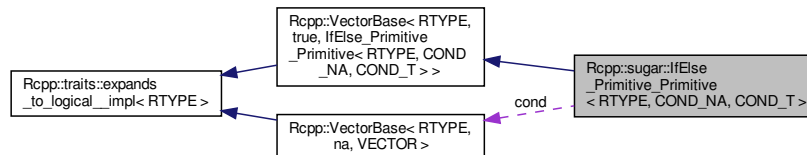
6.260 Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >:



Collaboration diagram for Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, COND\_NA, COND\_T > [COND\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)

## Public Member Functions

- [IfElse\\_Primitive\\_Primitive](#) (const [COND\\_TYPE](#) &cond\_, [STORAGE](#) lhs\_, [STORAGE](#) rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [COND\\_TYPE](#) & [cond](#)
- [STORAGE](#) [lhs](#)
- [STORAGE](#) [rhs](#)
- [STORAGE](#) [na](#)

### 6.260.1 Detailed Description

```
template<int RTYPE, bool COND_NA, typename COND_T>
class Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >
```

Definition at line 273 of file ifelse.h.

### 6.260.2 Member Typedef Documentation

#### 6.260.2.1 COND\_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T >
typedef Rcpp::VectorBase<LGLSXP, COND_NA, COND_T> Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE,
COND_NA, COND_T >::COND_TYPE
```

Definition at line 279 of file ifelse.h.

#### 6.260.2.2 STORAGE

```
template<int RTYPE, bool COND_NA, typename COND_T >
typedef traits::storage\_type<RTYPE>::type Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND_↔
NA, COND_T >::STORAGE
```

Definition at line 280 of file ifelse.h.

## 6.260.3 Constructor & Destructor Documentation

### 6.260.3.1 IfElse\_Primitive\_Primitive()

```
template<int RTYPE, bool COND_NA, typename COND_T >
Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::IfElse_Primitive_Primitive (
    const COND_TYPE & cond_,
    STORAGE lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 282 of file ifelse.h.

## 6.260.4 Member Function Documentation

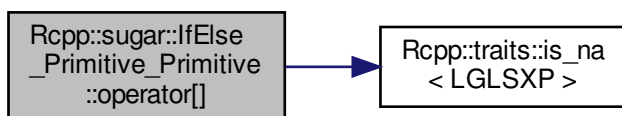
### 6.260.4.1 operator[]()

```
template<int RTYPE, bool COND_NA, typename COND_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 287 of file ifelse.h.

References Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::cond, Rcpp::traits::is\_na< LGLSXP >(), Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::lhs, and Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::rhs.

Here is the call graph for this function:



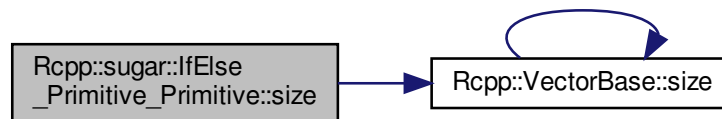
### 6.260.4.2 size()

```
template<int RTYPE, bool COND_NA, typename COND_T >
R_xlen_t Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::size ( ) const [inline]
```

Definition at line 293 of file ifelse.h.

References Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::cond, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



## 6.260.5 Member Data Documentation

### 6.260.5.1 cond

```
template<int RTYPE, bool COND_NA, typename COND_T >
const COND_TYPE& Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::cond [private]
```

Definition at line 296 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::operator[](), Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, false, COND\_T >::operator[](), Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::size(), and Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, false, COND\_T >::size().

### 6.260.5.2 lhs

```
template<int RTYPE, bool COND_NA, typename COND_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::lhs [private]
```

Definition at line 297 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::operator[](), and Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, false, COND\_T >::operator[]().

## 6.260.5.3 na

```
template<int RTYPE, bool COND_NA, typename COND_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::na [private]
```

Definition at line 299 of file ifelse.h.

## 6.260.5.4 rhs

```
template<int RTYPE, bool COND_NA, typename COND_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, COND_NA, COND_T >::rhs [private]
```

Definition at line 298 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::operator[()], and Rcpp←  
::sugar::IfElse\_Primitive\_Primitive< RTYPE, false, COND\_T >::operator[()].

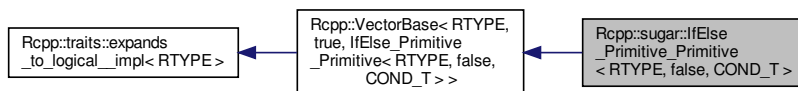
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

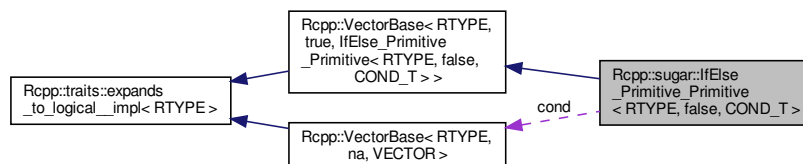
## 6.261 Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, false, COND\_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, false, COND\_T >:



Collaboration diagram for Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, false, COND\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, false, COND\_T > [COND\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)

## Public Member Functions

- [IfElse\\_Primitive\\_Primitive](#) (const [COND\\_TYPE](#) &cond\_, [STORAGE](#) lhs\_, [STORAGE](#) rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [COND\\_TYPE](#) & [cond](#)
- [STORAGE](#) lhs
- [STORAGE](#) rhs

### 6.261.1 Detailed Description

```
template<int RTYPE, typename COND_T>
class Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >
```

Definition at line 306 of file ifelse.h.

### 6.261.2 Member Typedef Documentation

#### 6.261.2.1 COND\_TYPE

```
template<int RTYPE, typename COND_T >
typedef Rcpp::VectorBase<LGLSXP, false, COND_T> Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE,
false, COND_T >::COND_TYPE
```

Definition at line 312 of file ifelse.h.

#### 6.261.2.2 STORAGE

```
template<int RTYPE, typename COND_T >
typedef traits::storage\_type<RTYPE>::type Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, false,
COND_T >::STORAGE
```

Definition at line 313 of file ifelse.h.

## 6.261.3 Constructor & Destructor Documentation

### 6.261.3.1 IfElse\_Primitive\_Primitive()

```
template<int RTYPE, typename COND_T >
Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::IfElse_Primitive_Primitive (
    const COND_TYPE & cond_,
    STORAGE lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 315 of file ifelse.h.

## 6.261.4 Member Function Documentation

### 6.261.4.1 operator[]()

```
template<int RTYPE, typename COND_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 320 of file ifelse.h.

References Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::cond, Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::lhs, and Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::rhs.

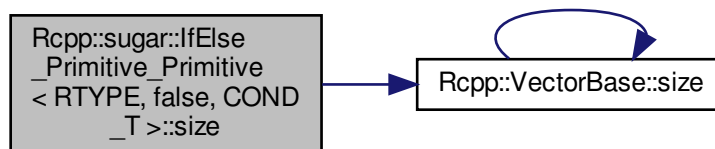
### 6.261.4.2 size()

```
template<int RTYPE, typename COND_T >
R_xlen_t Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::size ( ) const [inline]
```

Definition at line 324 of file ifelse.h.

References Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::cond, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



## 6.261.5 Member Data Documentation

### 6.261.5.1 cond

```
template<int RTYPE, typename COND_T >  
const COND_TYPE& Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::cond [private]
```

Definition at line 327 of file ifelse.h.

### 6.261.5.2 lhs

```
template<int RTYPE, typename COND_T >  
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::lhs [private]
```

Definition at line 328 of file ifelse.h.

### 6.261.5.3 rhs

```
template<int RTYPE, typename COND_T >  
STORAGE Rcpp::sugar::IfElse_Primitive_Primitive< RTYPE, false, COND_T >::rhs [private]
```

Definition at line 329 of file ifelse.h.

The documentation for this class was generated from the following file:

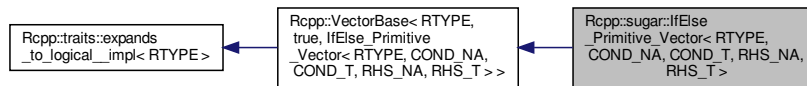
- [inst/include/Rcpp/sugar/functions/ifelse.h](#)



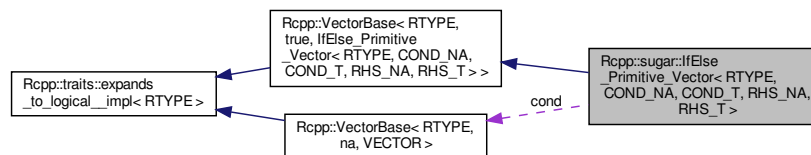
## 6.262 Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, COND\_NA, COND\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, COND\_NA, COND\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, COND\_NA, COND\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, COND\_NA, COND\_T > [COND\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE,RHS\_NA,RHS\_T > [RHS\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE,RHS\_NA,RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [IfElse\\_Primitive\\_Vector](#) (const [COND\\_TYPE](#) &cond\_, [STORAGE](#) lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [COND\\_TYPE](#) & [cond](#)
- [STORAGE](#) lhs
- const [RHS\\_EXT](#) & [rhs](#)

## 6.262.1 Detailed Description

```
template<int RTYPE, bool COND_NA, typename COND_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >
```

Definition at line 119 of file ifelse.h.

## 6.262.2 Member Typedef Documentation

### 6.262.2.1 COND\_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,COND_NA,COND_T> Rcpp::sugar::IfElse_Primitive_Vector< RTYPE,
COND_NA, COND_T, RHS_NA, RHS_T >::COND_TYPE
```

Definition at line 125 of file ifelse.h.

### 6.262.2.2 RHS\_EXT

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE ,RHS_NA ,RHS_T>::type Rcpp::sugar::IfElse_Primitive_Vector<
RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 129 of file ifelse.h.

### 6.262.2.3 RHS\_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE ,RHS_NA ,RHS_T> Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND↔
_NA, COND_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 126 of file ifelse.h.

### 6.262.2.4 STORAGE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA,
COND_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 127 of file ifelse.h.

## 6.262.3 Constructor & Destructor Documentation

### 6.262.3.1 IfElse\_Primitive\_Vector()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::IfElse_Primitive_Vector
(
    const COND_TYPE & cond_,
    STORAGE lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 131 of file ifelse.h.

## 6.262.4 Member Function Documentation

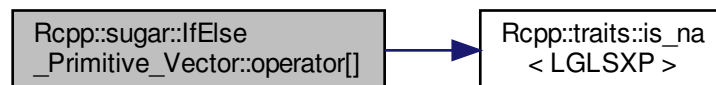
### 6.262.4.1 operator[]()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::operator[]
(
    R_xlen_t i ) const [inline]
```

Definition at line 136 of file ifelse.h.

References `Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::cond`, `Rcpp::traits::is_na< LGLSXP >()`, `Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::rhs`.

Here is the call graph for this function:



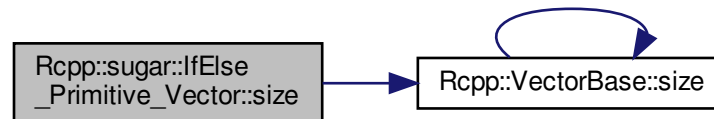
### 6.262.4.2 size()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::size ( )
const [inline]
```

Definition at line 143 of file ifelse.h.

References `Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::cond`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`.

Here is the call graph for this function:



## 6.262.5 Member Data Documentation

### 6.262.5.1 cond

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
const COND_TYPE& Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >↔
::cond [private]
```

Definition at line 146 of file ifelse.h.

Referenced by `Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::operator[]()`, `Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::operator[]()`, `Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::size()`, and `Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::size()`.

6.262.5.2 lhs

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 147 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, COND\_NA, COND\_T, RHS\_NA, RHS\_T >::operator[](), and Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, false, COND\_T, RHS\_NA, RHS\_T >::operator[]().

6.262.5.3 rhs

```
template<int RTYPE, bool COND_NA, typename COND_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::rhs [private]
```

Definition at line 148 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, COND\_NA, COND\_T, RHS\_NA, RHS\_T >::operator[](), and Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, false, COND\_T, RHS\_NA, RHS\_T >::operator[]().

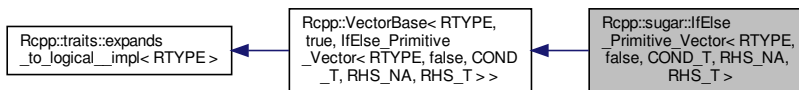
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/ifelse.h

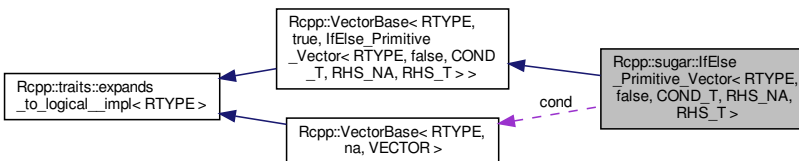
6.263 Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, false, COND\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, false, COND\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, false, COND\_T, RHS\_NA, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, false, COND\_T > [COND\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE,RHS\_NA,RHS\_T > [RHS\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE,RHS\_NA,RHS\_T >::type [RHS\\_EXT](#)

## Public Member Functions

- [IfElse\\_Primitive\\_Vector](#) (const [COND\\_TYPE](#) &cond\_, [STORAGE](#) lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [COND\\_TYPE](#) & [cond](#)
- [STORAGE](#) [lhs](#)
- const [RHS\\_EXT](#) & [rhs](#)

### 6.263.1 Detailed Description

```
template<int RTYPE, typename COND_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >
```

Definition at line 157 of file ifelse.h.

### 6.263.2 Member Typedef Documentation

#### 6.263.2.1 COND\_TYPE

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP,false,COND_T> Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, false,
COND_T, RHS_NA, RHS_T >::COND_TYPE
```

Definition at line 163 of file ifelse.h.

### 6.263.2.2 RHS\_EXT

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE ,RHS_NA ,RHS_T>::type Rcpp::sugar::IfElse_Primitive_Vector<
RTYPE, false, COND_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 166 of file ifelse.h.

### 6.263.2.3 RHS\_TYPE

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE ,RHS_NA ,RHS_T> Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false,
COND_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 164 of file ifelse.h.

### 6.263.2.4 STORAGE

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false,
COND_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 165 of file ifelse.h.

## 6.263.3 Constructor & Destructor Documentation

### 6.263.3.1 IfElse\_Primitive\_Vector()

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::IfElse_Primitive_Vector
(
    const COND_TYPE & cond_,
    STORAGE lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 168 of file ifelse.h.

### 6.263.4 Member Function Documentation

### 6.263.4.1 operator[]()

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 173 of file ifelse.h.

References `Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::cond`, `Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::rhs`.

### 6.263.4.2 size()

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::size ( )
const [inline]
```

Definition at line 178 of file ifelse.h.

References `Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >::cond`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`.

Here is the call graph for this function:



## 6.263.5 Member Data Documentation

### 6.263.5.1 cond

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
const COND_TYPE& Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::cond [private]
```

Definition at line 181 of file ifelse.h.



### 6.263.5.2 lhs

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 182 of file ifelse.h.

### 6.263.5.3 rhs

```
template<int RTYPE, typename COND_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::IfElse_Primitive_Vector< RTYPE, false, COND_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 183 of file ifelse.h.

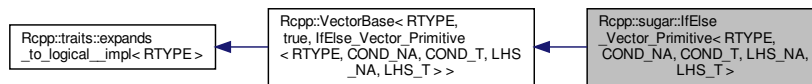
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/ifelse.h](#)

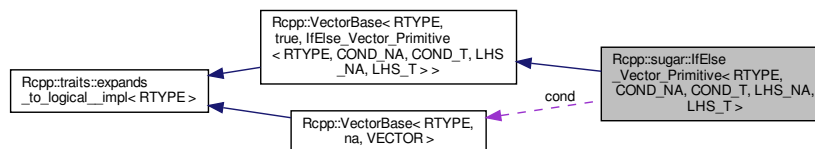
## 6.264 Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T >:



Collaboration diagram for Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, COND\_NA, COND\_T > [COND\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE,LHS\_NA,LHS\_T > [LHS\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE,LHS\_NA,LHS\_T >::type [LHS\\_EXT](#)

## Public Member Functions

- [IfElse\\_Vector\\_Primitive](#) (const [COND\\_TYPE](#) &cond\_, const [LHS\\_TYPE](#) &lhs\_, [STORAGE](#) rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [COND\\_TYPE](#) & [cond](#)
- const [LHS\\_EXT](#) & [lhs](#)
- const [STORAGE](#) [rhs](#)

### 6.264.1 Detailed Description

```
template<int RTYPE, bool COND_NA, typename COND_T, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >
```

Definition at line 196 of file ifelse.h.

### 6.264.2 Member Typedef Documentation

#### 6.264.2.1 COND\_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
typedef Rcpp::VectorBase<LGLSXP,COND_NA,COND_T> Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE,
COND_NA, COND_T, LHS_NA, LHS_T >::COND_TYPE
```

Definition at line 202 of file ifelse.h.

### 6.264.2.2 LHS\_EXT

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
typedef Rcpp::traits::Extractor<RTYPE ,LHS_NA ,LHS_T>::type Rcpp::sugar::IfElse_Vector_Primitive<
RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::LHS_EXT
```

Definition at line 205 of file ifelse.h.

### 6.264.2.3 LHS\_TYPE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
typedef Rcpp::VectorBase<RTYPE ,LHS_NA ,LHS_T> Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND↔
_NA, COND_T, LHS_NA, LHS_T >::LHS_TYPE
```

Definition at line 203 of file ifelse.h.

### 6.264.2.4 STORAGE

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA,
COND_T, LHS_NA, LHS_T >::STORAGE
```

Definition at line 204 of file ifelse.h.

## 6.264.3 Constructor & Destructor Documentation

### 6.264.3.1 IfElse\_Vector\_Primitive()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::IfElse_Vector_Primitive
(
    const COND_TYPE & cond_,
    const LHS_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 207 of file ifelse.h.

## 6.264.4 Member Function Documentation

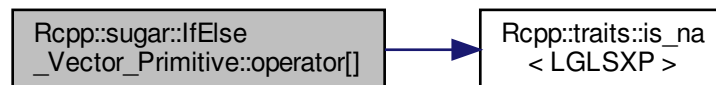
### 6.264.4.1 operator[]()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
STORAGE Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::operator[]
(
    R_xlen_t i ) const [inline]
```

Definition at line 212 of file ifelse.h.

References `Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::cond`, `Rcpp::traits::is_na< LGLSXP >()`, `Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::lhs`, and `Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::rhs`.

Here is the call graph for this function:



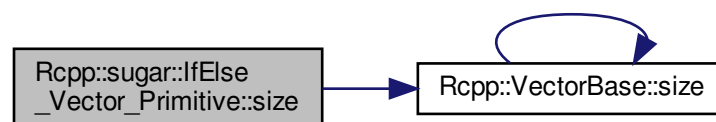
### 6.264.4.2 size()

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::size ( )
const [inline]
```

Definition at line 219 of file ifelse.h.

References `Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::cond`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`.

Here is the call graph for this function:



## 6.264.5 Member Data Documentation

### 6.264.5.1 cond

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
const COND_TYPE& Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::
::cond [private]
```

Definition at line 222 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T >::operator[](), Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, false, COND\_T, LHS\_NA, LHS\_T >::operator[](), Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T >::size(), and Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, false, COND\_T, LHS\_NA, LHS\_T >::size().

### 6.264.5.2 lhs

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
const LHS_EXT& Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::lhs
[private]
```

Definition at line 223 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T >::operator[](), and Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, false, COND\_T, LHS\_NA, LHS\_T >::operator[]().

### 6.264.5.3 rhs

```
template<int RTYPE, bool COND_NA, typename COND_T , bool LHS_NA, typename LHS_T >
const STORAGE Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T >::rhs
[private]
```

Definition at line 224 of file ifelse.h.

Referenced by Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T >::operator[](), and Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, false, COND\_T, LHS\_NA, LHS\_T >::operator[]().

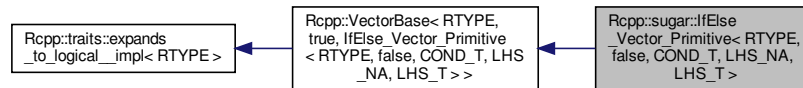
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/ifelse.h](#)

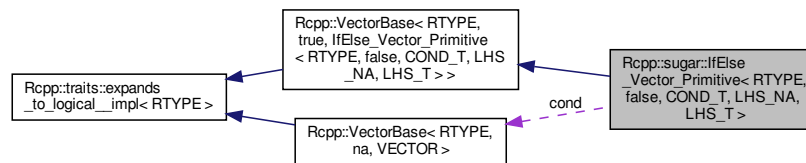
## 6.265 Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, false, COND\_T, LHS\_NA, LHS\_T > Class Template Reference

```
#include <ifelse.h>
```

Inheritance diagram for Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, false, COND\_T, LHS\_NA, LHS\_T >:



Collaboration diagram for Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, false, COND\_T, LHS\_NA, LHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, false, COND\_T > [COND\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE,LHS\_NA,LHS\_T > [LHS\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE,LHS\_NA,LHS\_T >::type [LHS\\_EXT](#)

### Public Member Functions

- [IfElse\\_Vector\\_Primitive](#) (const [COND\\_TYPE](#) &cond\_, const [LHS\\_TYPE](#) &lhs\_, [STORAGE](#) rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [COND\\_TYPE](#) & [cond](#)
- const [LHS\\_EXT](#) & [lhs](#)
- const [STORAGE](#) [rhs](#)

## 6.265.1 Detailed Description

```
template<int RTYPE, typename COND_T, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >
```

Definition at line 233 of file ifelse.h.

## 6.265.2 Member Typedef Documentation

### 6.265.2.1 COND\_TYPE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
typedef Rcpp::VectorBase<LGLSXP, false, COND_T> Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false,
COND_T, LHS_NA, LHS_T >::COND_TYPE
```

Definition at line 239 of file ifelse.h.

### 6.265.2.2 LHS\_EXT

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
typedef Rcpp::traits::Extractor<RTYPE ,LHS_NA ,LHS_T>::type Rcpp::sugar::IfElse_Vector_Primitive<
RTYPE, false, COND_T, LHS_NA, LHS_T >::LHS_EXT
```

Definition at line 242 of file ifelse.h.

### 6.265.2.3 LHS\_TYPE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
typedef Rcpp::VectorBase<RTYPE ,LHS_NA ,LHS_T> Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false,
COND_T, LHS_NA, LHS_T >::LHS_TYPE
```

Definition at line 240 of file ifelse.h.

### 6.265.2.4 STORAGE

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false,
COND_T, LHS_NA, LHS_T >::STORAGE
```

Definition at line 241 of file ifelse.h.

## 6.265.3 Constructor & Destructor Documentation

### 6.265.3.1 IfElse\_Vector\_Primitive()

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::IfElse_Vector_Primitive
(
    const COND_TYPE & cond_,
    const LHS_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 244 of file ifelse.h.

## 6.265.4 Member Function Documentation

### 6.265.4.1 operator[]()

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
STORAGE Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 249 of file ifelse.h.

References [Rcpp::sugar::IfElse\\_Vector\\_Primitive< RTYPE, COND\\_NA, COND\\_T, LHS\\_NA, LHS\\_T >::cond](#), [Rcpp::sugar::IfElse\\_Vector\\_Primitive< RTYPE, COND\\_NA, COND\\_T, LHS\\_NA, LHS\\_T >::lhs](#), and [Rcpp::sugar::IfElse\\_Vector\\_Primitive< RTYPE, COND\\_NA, COND\\_T, LHS\\_NA, LHS\\_T >::rhs](#).



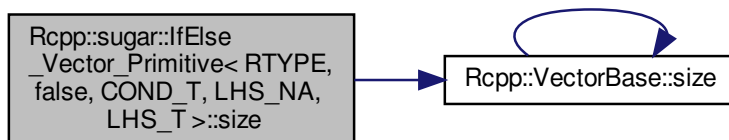
### 6.265.4.2 size()

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::size ( )
const [inline]
```

Definition at line 254 of file ifelse.h.

References Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T >::cond, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



## 6.265.5 Member Data Documentation

### 6.265.5.1 cond

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
const COND_TYPE& Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >←
::cond [private]
```

Definition at line 257 of file ifelse.h.

### 6.265.5.2 lhs

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
const LHS_EXT& Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::lhs
[private]
```

Definition at line 258 of file ifelse.h.

### 6.265.5.3 rhs

```
template<int RTYPE, typename COND_T , bool LHS_NA, typename LHS_T >
const STORAGE Rcpp::sugar::IfElse_Vector_Primitive< RTYPE, false, COND_T, LHS_NA, LHS_T >::rhs
[private]
```

Definition at line 259 of file ifelse.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/ifelse.h](#)

## 6.266 Rcpp::sugar::In< RTYPE, TABLE\_T > Class Template Reference

```
#include <unique.h>
```

### Public Member Functions

- [In](#) (const TABLE\_T &table)
- [template<typename T > LogicalVector get](#) (const T &x) const

### Private Types

- [typedef sugar::IndexHash< RTYPE > HASH](#)

### Private Attributes

- [Vector< RTYPE > vec](#)
- [HASH hash](#)

### 6.266.1 Detailed Description

```
template<int RTYPE, typename TABLE_T>
class Rcpp::sugar::In< RTYPE, TABLE_T >
```

Definition at line 44 of file unique.h.

### 6.266.2 Member Typedef Documentation

### 6.266.2.1 HASH

```
template<int RTYPE, typename TABLE_T >
typedef sugar::IndexHash<RTYPE> Rcpp::sugar::In< RTYPE, TABLE_T >::HASH [private]
```

Definition at line 46 of file unique.h.

## 6.266.3 Constructor & Destructor Documentation

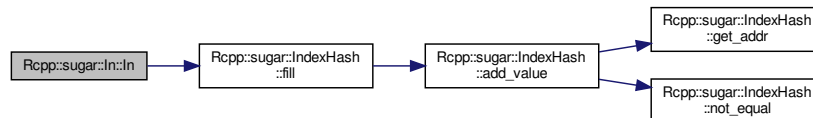
### 6.266.3.1 In()

```
template<int RTYPE, typename TABLE_T >
Rcpp::sugar::In< RTYPE, TABLE_T >::In (
    const TABLE_T & table ) [inline]
```

Definition at line 50 of file unique.h.

References [Rcpp::sugar::IndexHash](#)< RTYPE >::fill(), and [Rcpp::sugar::In](#)< RTYPE, TABLE\_T >::hash.

Here is the call graph for this function:



## 6.266.4 Member Function Documentation

### 6.266.4.1 get()

```
template<int RTYPE, typename TABLE_T >
template<typename T >
LogicalVector Rcpp::sugar::In< RTYPE, TABLE_T >::get (
    const T & x ) const [inline]
```

Definition at line 55 of file unique.h.

References [Rcpp::sugar::In](#)< RTYPE, TABLE\_T >::hash.

Referenced by [Rcpp::in](#)().

## 6.266.5 Member Data Documentation

### 6.266.5.1 hash

```
template<int RTYPE, typename TABLE_T >  
HASH Rcpp::sugar::In< RTYPE, TABLE_T >::hash [private]
```

Definition at line 47 of file unique.h.

Referenced by `Rcpp::sugar::In< RTYPE, TABLE_T >::get()`, and `Rcpp::sugar::In< RTYPE, TABLE_T >::In()`.

### 6.266.5.2 vec

```
template<int RTYPE, typename TABLE_T >  
Vector<RTYPE> Rcpp::sugar::In< RTYPE, TABLE_T >::vec [private]
```

Definition at line 45 of file unique.h.

The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/unique.h`

## 6.267 Rcpp::sugar::IndexHash< RTYPE > Class Template Reference

```
#include <IndexHash.h>
```

### Public Types

- typedef `traits::storage_type< RTYPE >::type` STORAGE
- typedef `Vector< RTYPE >` VECTOR

## Public Member Functions

- [IndexHash](#) (SEXP [table](#))
- [IndexHash](#) & [fill](#) ()
- [LogicalVector](#) [fill\\_and\\_get\\_duplicated](#) ()
- `template<typename T >`  
SEXP [lookup](#) (const T &vec) const
- SEXP [lookup](#) (const [VECTOR](#) &vec) const
- bool [contains](#) ([STORAGE](#) val) const
- int [size](#) () const
- [Vector](#)< RTYPE > [keys](#) () const
- `template<typename T >`  
SEXP [lookup\\_\\_impl](#) (const T &vec, int n\_) const
- SEXP [get\\_profile\\_data](#) ()
- bool [not\\_equal](#) (const [STORAGE](#) &lhs, const [STORAGE](#) &rhs)
- bool [add\\_value](#) (int i)
- uint32\_t [get\\_index](#) ([STORAGE](#) value) const
- uint32\_t [get\\_addr](#) ([STORAGE](#) value) const
- uint32\_t [get\\_addr](#) (int value) const
- uint32\_t [get\\_addr](#) (double val) const
- uint32\_t [get\\_addr](#) (SEXP value) const

## Public Attributes

- int [n](#)
- int [m](#)
- int [k](#)
- [STORAGE](#) \* [src](#)
- int [size\\_](#)
- int \* [data](#)

### 6.267.1 Detailed Description

```
template<int RTYPE>
class Rcpp::sugar::IndexHash< RTYPE >
```

Definition at line 49 of file `IndexHash.h`.

### 6.267.2 Member Typedef Documentation

### 6.267.2.1 STORAGE

```
template<int RTYPE>
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IndexHash< RTYPE >::STORAGE
```

Definition at line 51 of file IndexHash.h.

### 6.267.2.2 VECTOR

```
template<int RTYPE>
typedef Vector<RTYPE> Rcpp::sugar::IndexHash< RTYPE >::VECTOR
```

Definition at line 52 of file IndexHash.h.

## 6.267.3 Constructor & Destructor Documentation

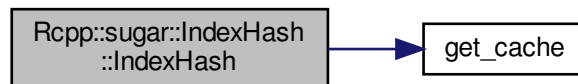
### 6.267.3.1 IndexHash()

```
template<int RTYPE>
Rcpp::sugar::IndexHash< RTYPE >::IndexHash (
    SEXP table ) [inline]
```

Definition at line 54 of file IndexHash.h.

References `Rcpp::sugar::IndexHash< RTYPE >::data`, `get_cache()`, `Rcpp::sugar::IndexHash< RTYPE >::k`, `Rcpp::sugar::IndexHash< RTYPE >::m`, `Rcpp::sugar::IndexHash< RTYPE >::n`, `RCPP_PROFILE_RECORD`, `RCPP_PROFILE_TIC`, and `RCPP_PROFILE_TOC`.

Here is the call graph for this function:



### 6.267.4 Member Function Documentation

### 6.267.4.1 add\_value()

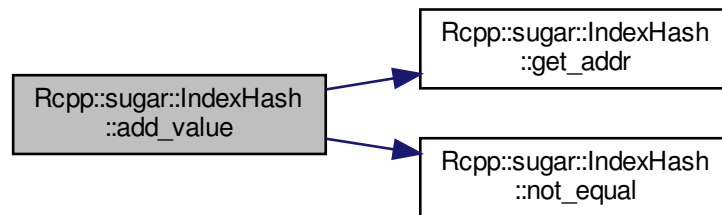
```
template<int RTYPE>
bool Rcpp::sugar::IndexHash< RTYPE >::add_value (
    int i ) [inline]
```

Definition at line 166 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::data, DEMANGLE, Rcpp::sugar::IndexHash< RTYPE >::get\_addr(), Rcpp::sugar::IndexHash< RTYPE >::m, Rcpp::sugar::IndexHash< RTYPE >::not\_equal(), RCPP\_DEBUG\_2, Rcpp::sugar::IndexHash< RTYPE >::size\_, and Rcpp::sugar::IndexHash< RTYPE >::src.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::fill(), and Rcpp::sugar::IndexHash< RTYPE >::fill\_and\_get\_duplicates().

Here is the call graph for this function:



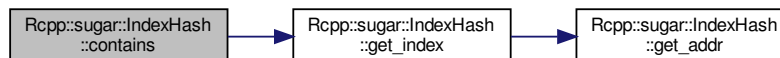
### 6.267.4.2 contains()

```
template<int RTYPE>
bool Rcpp::sugar::IndexHash< RTYPE >::contains (
    STORAGE val ) const [inline]
```

Definition at line 101 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::get\_index().

Here is the call graph for this function:



### 6.267.4.3 fill()

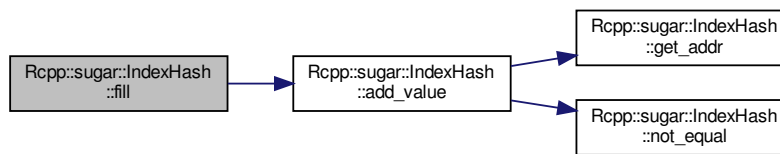
```
template<int RTYPE>
IndexHash& Rcpp::sugar::IndexHash< RTYPE >::fill ( ) [inline]
```

Definition at line 73 of file IndexHash.h.

References `Rcpp::sugar::IndexHash< RTYPE >::add_value()`, `Rcpp::sugar::IndexHash< RTYPE >::n`, `Rcpp::PROFILE_RECORD`, `Rcpp::PROFILE_TIC`, and `Rcpp::PROFILE_TOC`.

Referenced by `Rcpp::sugar::In< RTYPE, TABLE_T >::In()`, `Rcpp::match()`, and `Rcpp::unique()`.

Here is the call graph for this function:



### 6.267.4.4 fill\_and\_get\_duplicated()

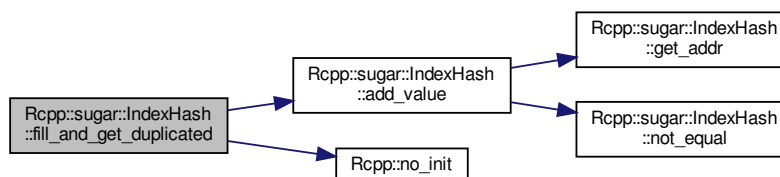
```
template<int RTYPE>
LogicalVector Rcpp::sugar::IndexHash< RTYPE >::fill_and_get_duplicated ( ) [inline]
```

Definition at line 84 of file IndexHash.h.

References `Rcpp::sugar::IndexHash< RTYPE >::add_value()`, `Rcpp::sugar::IndexHash< RTYPE >::n`, and `Rcpp::no_init()`.

Referenced by `Rcpp::duplicated()`.

Here is the call graph for this function:





**6.267.4.5 get\_addr() [1/4]**

```
uint32_t Rcpp::sugar::IndexHash< REALSXP >::get_addr (
    double val ) const [inline]
```

Definition at line 207 of file IndexHash.h.

References RCPP\_HASH.

**6.267.4.6 get\_addr() [2/4]**

```
uint32_t Rcpp::sugar::IndexHash< INTSXP >::get_addr (
    int value ) const [inline]
```

Definition at line 203 of file IndexHash.h.

References RCPP\_HASH.

**6.267.4.7 get\_addr() [3/4]**

```
uint32_t Rcpp::sugar::IndexHash< STRSXP >::get_addr (
    SEXP value ) const [inline]
```

Definition at line 224 of file IndexHash.h.

References RCPP\_HASH.

**6.267.4.8 get\_addr() [4/4]**

```
template<int RTYPE>
uint32_t Rcpp::sugar::IndexHash< RTYPE >::get_addr (
    STORAGE value ) const
```

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add\_value(), and Rcpp::sugar::IndexHash< RTYPE >::get\_↵  
index().

### 6.267.4.9 get\_index()

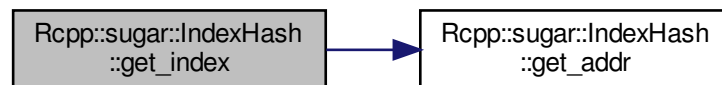
```
template<int RTYPE>
uint32_t Rcpp::sugar::IndexHash< RTYPE >::get_index (
    STORAGE value ) const [inline]
```

Definition at line 187 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::data, Rcpp::sugar::IndexHash< RTYPE >::get\_addr(), Rcpp::sugar::IndexHash< RTYPE >::m, and Rcpp::sugar::IndexHash< RTYPE >::src.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::contains(), and Rcpp::sugar::IndexHash< RTYPE >::lookup\_\_impl().

Here is the call graph for this function:



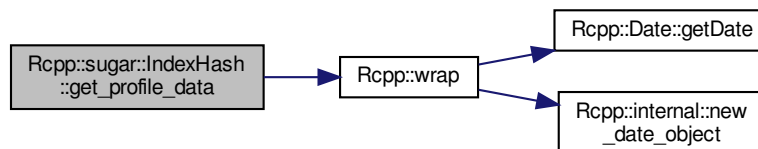
### 6.267.4.10 get\_profile\_data()

```
template<int RTYPE>
SEXP Rcpp::sugar::IndexHash< RTYPE >::get_profile_data ( ) [inline]
```

Definition at line 154 of file IndexHash.h.

References Rcpp::wrap().

Here is the call graph for this function:



**6.267.4.11 keys()**

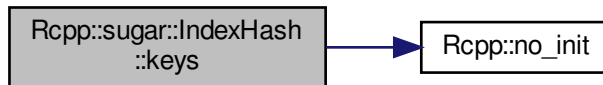
```
template<int RTYPE>
Vector<RTYPE> Rcpp::sugar::IndexHash< RTYPE >::keys ( ) const [inline]
```

Definition at line 110 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::data, Rcpp::no\_init(), Rcpp::sugar::IndexHash< RTYPE >::size\_←, and Rcpp::sugar::IndexHash< RTYPE >::src.

Referenced by Rcpp::unique().

Here is the call graph for this function:

**6.267.4.12 lookup() [1/2]**

```
template<int RTYPE>
template<typename T >
SEXP Rcpp::sugar::IndexHash< RTYPE >::lookup (
    const T & vec ) const [inline]
```

Definition at line 92 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::lookup\_\_impl().

Referenced by Rcpp::sugar::IndexHash< RTYPE >::lookup\_\_impl(), and Rcpp::match().

Here is the call graph for this function:



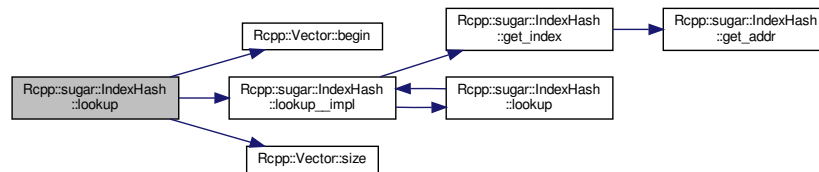
### 6.267.4.13 lookup() [2/2]

```
template<int RTYPE>
SEXP Rcpp::sugar::IndexHash< RTYPE >::lookup (
    const VECTOR & vec ) const [inline]
```

Definition at line 97 of file IndexHash.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::sugar::IndexHash< RTYPE >::lookup\_\_impl(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



### 6.267.4.14 lookup\_\_impl()

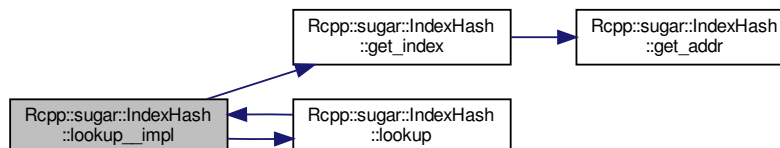
```
template<int RTYPE>
template<typename T >
SEXP Rcpp::sugar::IndexHash< RTYPE >::lookup__impl (
    const T & vec,
    int n_ ) const [inline]
```

Definition at line 134 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::get\_index(), Rcpp::sugar::IndexHash< RTYPE >::lookup(), RCPP←\_PROFILE\_RECORD, RCPP\_PROFILE\_TIC, and RCPP\_PROFILE\_TOC.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::lookup().

Here is the call graph for this function:



#### 6.267.4.15 not\_equal()

```
template<int RTYPE>
bool Rcpp::sugar::IndexHash< RTYPE >::not_equal (
    const STORAGE & lhs,
    const STORAGE & rhs ) [inline]
```

Definition at line 162 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add\_value().

#### 6.267.4.16 size()

```
template<int RTYPE>
int Rcpp::sugar::IndexHash< RTYPE >::size ( ) const [inline]
```

Definition at line 105 of file IndexHash.h.

References Rcpp::sugar::IndexHash< RTYPE >::size\_.

### 6.267.5 Member Data Documentation

#### 6.267.5.1 data

```
template<int RTYPE>
int* Rcpp::sugar::IndexHash< RTYPE >::data
```

Definition at line 122 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add\_value(), Rcpp::sugar::IndexHash< RTYPE >::get\_index(), Rcpp::sugar::IndexHash< RTYPE >::IndexHash(), and Rcpp::sugar::IndexHash< RTYPE >::keys().

#### 6.267.5.2 k

```
template<int RTYPE>
int Rcpp::sugar::IndexHash< RTYPE >::k
```

Definition at line 118 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::IndexHash().

### 6.267.5.3 m

```
template<int RTYPE>
int Rcpp::sugar::IndexHash< RTYPE >::m
```

Definition at line 118 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add\_value(), Rcpp::sugar::IndexHash< RTYPE >::get\_index(), and Rcpp::sugar::IndexHash< RTYPE >::IndexHash().

### 6.267.5.4 n

```
template<int RTYPE>
int Rcpp::sugar::IndexHash< RTYPE >::n
```

Definition at line 118 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::fill(), Rcpp::sugar::IndexHash< RTYPE >::fill\_and\_get\_↔ duplicated(), and Rcpp::sugar::IndexHash< RTYPE >::IndexHash().

### 6.267.5.5 size\_

```
template<int RTYPE>
int Rcpp::sugar::IndexHash< RTYPE >::size_
```

Definition at line 120 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add\_value(), Rcpp::sugar::IndexHash< RTYPE >::keys(), and Rcpp::sugar::IndexHash< RTYPE >::size().

### 6.267.5.6 src

```
template<int RTYPE>
STORAGE* Rcpp::sugar::IndexHash< RTYPE >::src
```

Definition at line 119 of file IndexHash.h.

Referenced by Rcpp::sugar::IndexHash< RTYPE >::add\_value(), Rcpp::sugar::IndexHash< RTYPE >::get\_index(), and Rcpp::sugar::IndexHash< RTYPE >::keys().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/hash/IndexHash.h

## 6.268 Rcpp::traits::init\_type< RTYPE > Struct Template Reference

```
#include <init_type.h>
```

### Public Types

- typedef [storage\\_type](#)< RTYPE >::type type

### 6.268.1 Detailed Description

```
template<int RTYPE>  
struct Rcpp::traits::init_type< RTYPE >
```

Definition at line 28 of file `init_type.h`.

### 6.268.2 Member Typedef Documentation

#### 6.268.2.1 type

```
template<int RTYPE>  
typedef storage\_type<RTYPE>::type Rcpp::traits::init_type< RTYPE >::type
```

Definition at line 29 of file `init_type.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/init_type.h`

## 6.269 Rcpp::traits::init\_type< LGLSXP > Struct Reference

```
#include <init_type.h>
```

### Public Types

- typedef bool [type](#)

### 6.269.1 Detailed Description

Definition at line 34 of file `init_type.h`.

### 6.269.2 Member Typedef Documentation

#### 6.269.2.1 type

```
typedef bool Rcpp::traits::init\_type< LGLSXP >::type
```

Definition at line 35 of file `init_type.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/init\\_type.h](#)

## 6.270 [Rcpp::traits::init\\_type](#)< STRSXP > Struct Reference

```
#include <init_type.h>
```

### Public Types

- typedef const char \* [type](#)

### 6.270.1 Detailed Description

Definition at line 31 of file `init_type.h`.

### 6.270.2 Member Typedef Documentation

#### 6.270.2.1 type

```
typedef const char* Rcpp::traits::init\_type< STRSXP >::type
```

Definition at line 32 of file `init_type.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/init\\_type.h](#)



## 6.271 Rcpp::traits::input\_parameter< T > Struct Template Reference

```
#include <InputParameter.h>
```

### Public Types

- typedef [Rcpp::InputParameter< T >](#) type

### 6.271.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::input_parameter< T >
```

Definition at line 82 of file InputParameter.h.

### 6.271.2 Member Typedef Documentation

#### 6.271.2.1 type

```
template<typename T >  
typedef Rcpp::InputParameter<T> Rcpp::traits::input\_parameter< T >::type
```

Definition at line 83 of file InputParameter.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/InputParameter.h](#)

## 6.272 Rcpp::traits::input\_parameter< const T & > Struct Template Reference

```
#include <InputParameter.h>
```

### Public Types

- typedef [Rcpp::ConstReferenceInputParameter< T >](#) type

### 6.272.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::input_parameter< const T & >
```

Definition at line 94 of file InputParameter.h.

### 6.272.2 Member Typedef Documentation

#### 6.272.2.1 type

```
template<typename T >
typedef Rcpp::ConstReferenceInputParameter<T> Rcpp::traits::input_parameter< const T & >::type
```

Definition at line 95 of file InputParameter.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/InputParameter.h](#)

## 6.273 Rcpp::traits::input\_parameter< const T > Struct Template Reference

```
#include <InputParameter.h>
```

### Public Types

- typedef [Rcpp::ConstInputParameter< T >](#) [type](#)

### 6.273.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::input_parameter< const T >
```

Definition at line 90 of file InputParameter.h.

### 6.273.2 Member Typedef Documentation

### 6.273.2.1 type

```
template<typename T >
typedef Rcpp::ConstInputParameter<T> Rcpp::traits::input_parameter< const T >::type
```

Definition at line 91 of file InputParameter.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/InputParameter.h](#)

## 6.274 Rcpp::traits::input\_parameter< T & > Struct Template Reference

```
#include <InputParameter.h>
```

### Public Types

- typedef [Rcpp::ReferenceInputParameter< T >](#) type

### 6.274.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::input_parameter< T & >
```

Definition at line 86 of file InputParameter.h.

### 6.274.2 Member Typedef Documentation

#### 6.274.2.1 type

```
template<typename T >
typedef Rcpp::ReferenceInputParameter<T> Rcpp::traits::input_parameter< T & >::type
```

Definition at line 87 of file InputParameter.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/InputParameter.h](#)

## 6.275 Rcpp::InputParameter< T > Class Template Reference

```
#include <InputParameter.h>
```

### Public Member Functions

- [InputParameter](#) (SEXP x\_)
- [operator T](#) ()

### Private Attributes

- [SEXP x](#)

### 6.275.1 Detailed Description

```
template<typename T>  
class Rcpp::InputParameter< T >
```

Definition at line 30 of file InputParameter.h.

### 6.275.2 Constructor & Destructor Documentation

#### 6.275.2.1 InputParameter()

```
template<typename T >  
Rcpp::InputParameter< T >::InputParameter (  
    SEXP x_ ) [inline]
```

Definition at line 32 of file InputParameter.h.

### 6.275.3 Member Function Documentation

#### 6.275.3.1 operator T()

```
template<typename T >  
Rcpp::InputParameter< T >::operator T ( ) [inline]
```

Definition at line 34 of file InputParameter.h.

References [Rcpp::InputParameter< T >::x](#).

## 6.275.4 Member Data Documentation

### 6.275.4.1 x

```
template<typename T >  
SEXPR Rcpp::InputParameter< T >::x [private]
```

Definition at line 37 of file InputParameter.h.

Referenced by Rcpp::InputParameter< T >::operator T().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/[InputParameter.h](#)

## 6.276 Rcpp::sugar::InSet< HASH > Class Template Reference

```
#include <unique.h>
```

### Public Member Functions

- [InSet](#) (const HASH &hash\_)
- int [operator\(\)](#) (STORAGE value)

### Private Types

- typedef HASH::STORAGE [STORAGE](#)

### Private Attributes

- const HASH & [hash](#)

### 6.276.1 Detailed Description

```
template<typename HASH>  
class Rcpp::sugar::InSet< HASH >
```

Definition at line 29 of file unique.h.

## 6.276.2 Member Typedef Documentation

### 6.276.2.1 STORAGE

```
template<typename HASH >
typedef HASH::STORAGE Rcpp::sugar::InSet< HASH >::STORAGE [private]
```

Definition at line 30 of file unique.h.

## 6.276.3 Constructor & Destructor Documentation

### 6.276.3.1 InSet()

```
template<typename HASH >
Rcpp::sugar::InSet< HASH >::InSet (
    const HASH & hash_ ) [inline]
```

Definition at line 33 of file unique.h.

## 6.276.4 Member Function Documentation

### 6.276.4.1 operator>()

```
template<typename HASH >
int Rcpp::sugar::InSet< HASH >::operator() (
    STORAGE value ) [inline]
```

Definition at line 35 of file unique.h.

References Rcpp::sugar::InSet< HASH >::hash.

## 6.276.5 Member Data Documentation

### 6.276.5.1 hash

```
template<typename HASH >  
const HASH& Rcpp::sugar::InSet< HASH >::hash [private]
```

Definition at line 40 of file unique.h.

Referenced by Rcpp::sugar::InSet< HASH >::operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/unique.h

## 6.277 Rcpp::traits::int2type< I > Struct Template Reference

```
#include <traits.h>
```

### Public Types

- enum { value = I }

### 6.277.1 Detailed Description

```
template<int I>  
struct Rcpp::traits::int2type< I >
```

Definition at line 33 of file traits.h.

### 6.277.2 Member Enumeration Documentation

#### 6.277.2.1 anonymous enum

```
template<int I>  
anonymous enum
```

#### Enumerator

value	
-------	--

Definition at line 33 of file traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/traits.h](#)

## 6.278 Rcpp::traits::integral\_constant< \_T, \_V > Struct Template Reference

```
#include <integral_constant.h>
```

Inherited by [Rcpp::traits::same\\_type< r\\_type\\_traits< T >::r\\_category, r\\_type\\_primitive\\_tag >](#), [Rcpp::traits::same\\_type< T::value\\_type, value\\_type >](#), [Rcpp::Matrix< RTYPE, StoragePolicy >::can\\_have\\_na](#), [Rcpp::sugar::is\\_sugar\\_vector< T >](#), [Rcpp::sugar::is\\_sugar\\_vector< Rcpp::Vector< RTYPE > >](#), [Rcpp::traits::has\\_na< RTYPE >](#), [Rcpp::traits::has\\_na< CPLXSXP >](#), [Rcpp::traits::has\\_na< INTSXP >](#), [Rcpp::traits::has\\_na< LGLSXP >](#), [Rcpp::traits::has\\_na< REALSXP >](#), [Rcpp::traits::has\\_na< STRSXP >](#), [Rcpp::traits::is\\_arithmetic< typename >](#), [Rcpp::traits::is\\_arithmetic< const double >](#), [Rcpp::traits::is\\_arithmetic< const float >](#), [Rcpp::traits::is\\_arithmetic< const int >](#), [Rcpp::traits::is\\_arithmetic< const long >](#), [Rcpp::traits::is\\_arithmetic< const long double >](#), [Rcpp::traits::is\\_arithmetic< const short >](#), [Rcpp::traits::is\\_arithmetic< const unsigned int >](#), [Rcpp::traits::is\\_arithmetic< const unsigned long >](#), [Rcpp::traits::is\\_arithmetic< const unsigned long long >](#), [Rcpp::traits::is\\_arithmetic< double >](#), [Rcpp::traits::is\\_arithmetic< float >](#), [Rcpp::traits::is\\_arithmetic< int >](#), [Rcpp::traits::is\\_arithmetic< long >](#), [Rcpp::traits::is\\_arithmetic< long double >](#), [Rcpp::traits::is\\_arithmetic< short >](#), [Rcpp::traits::is\\_arithmetic< unsigned int >](#), [Rcpp::traits::is\\_arithmetic< unsigned long >](#), [Rcpp::traits::is\\_arithmetic< unsigned short >](#), [Rcpp::traits::is\\_bool< typename >](#), [Rcpp::traits::is\\_bool< bool >](#), [Rcpp::traits::is\\_bool< const bool >](#), [Rcpp::traits::is\\_bool< volatile bool >](#), [Rcpp::traits::is\\_const< typename >](#), [Rcpp::traits::is\\_const< \\_Tp const >](#), [Rcpp::traits::is\\_named< T >](#), [Rcpp::traits::is\\_named< Rcpp::Argument >](#), [Rcpp::traits::is\\_named< named\\_object< T > >](#), [Rcpp::traits::is\\_pointer< T >](#), [Rcpp::traits::is\\_pointer< T \\* >](#), [Rcpp::traits::is\\_reference< typename >](#), [Rcpp::traits::is\\_reference< \\_Tp & >](#), [Rcpp::traits::is\\_trivial< RTYPE >](#), [Rcpp::traits::is\\_trivial< EXPRSXP >](#), [Rcpp::traits::is\\_trivial< VECSXP >](#), [Rcpp::traits::is\\_wide\\_string< char >](#), [Rcpp::traits::is\\_wide\\_string< const char \\* >](#), [Rcpp::traits::is\\_wide\\_string< const wchar\\_t \\* >](#), [Rcpp::traits::is\\_wide\\_string< wchar\\_t >](#), [Rcpp::traits::needs\\_protection< T >](#), [Rcpp::traits::needs\\_protection< SEXP >](#), [Rcpp::traits::r\\_sexptype\\_needs cast< T >](#), [Rcpp::traits::r\\_sexptype\\_needs cast< Rbyte >](#), [Rcpp::traits::r\\_sexptype\\_needs cast< Rcomplex >](#), [Rcpp::traits::r\\_sexptype\\_needs cast< Rcpp::traits::r\\_sexptype\\_needs cast< int >](#), [Rcpp::traits::same\\_type< T, U >](#), and [Rcpp::traits::same\\_type< T, T >](#).

### Public Types

- typedef [\\_T value\\_type](#)
- typedef [integral\\_constant< \\_T, \\_V > type](#)

### Static Public Attributes

- static const [\\_T value](#) = [\\_V](#)

#### 6.278.1 Detailed Description

```
template<typename _T, _T _V>
struct Rcpp::traits::integral_constant< _T, _V >
```

Definition at line 29 of file [integral\\_constant.h](#).



## 6.278.2 Member Typedef Documentation

### 6.278.2.1 type

```
template<typename _T , _T _V>  
typedef integral\_constant<\_T,\_V> Rcpp::traits::integral\_constant<\_T, \_V >::type
```

Definition at line 32 of file `integral_constant.h`.

### 6.278.2.2 value\_type

```
template<typename _T , _T _V>  
typedef _T Rcpp::traits::integral\_constant<\_T, \_V >::value\_type
```

Definition at line 31 of file `integral_constant.h`.

## 6.278.3 Member Data Documentation

### 6.278.3.1 value

```
template<typename _T , _T _V>  
const _T Rcpp::traits::integral\_constant<\_T, \_V >::value = _V [static]
```

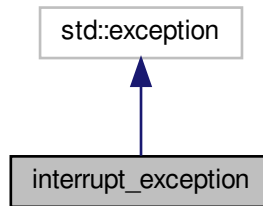
Definition at line 30 of file `integral_constant.h`.

The documentation for this struct was generated from the following file:

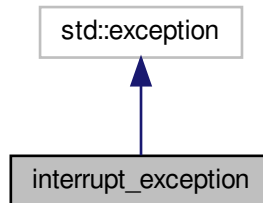
- [inst/include/Rcpp/traits/integral\\_constant.h](#)

## 6.279 interrupt\_exception Class Reference

Inheritance diagram for interrupt\_exception:



Collaboration diagram for interrupt\_exception:



### Public Member Functions

- [interrupt\\_exception](#) (std::string message)
- virtual [~interrupt\\_exception](#) () throw ()
- virtual const char \* [what](#) () const throw ()

### Public Attributes

- std::string [detailed\\_message](#)

### 6.279.1 Detailed Description

Base class for interrupt exceptions thrown when user interrupts are detected.

Definition at line 15 of file `piWithInterrupts.cpp`.

## 6.279.2 Constructor & Destructor Documentation

### 6.279.2.1 interrupt\_exception()

```
interrupt_exception::interrupt_exception (
    std::string message ) [inline]
```

Constructor.

#### Parameters

in	<i>message</i>	A description of event that caused this exception.
----	----------------	--

Definition at line 22 of file piWithInterrupts.cpp.

### 6.279.2.2 ~interrupt\_exception()

```
virtual interrupt_exception::~interrupt_exception ( ) throw ( ) [inline], [virtual]
```

Virtual destructor. Needed to avoid "looser throw specification" errors.

Definition at line 29 of file piWithInterrupts.cpp.

## 6.279.3 Member Function Documentation

### 6.279.3.1 what()

```
virtual const char* interrupt_exception::what ( ) const throw ( ) [inline], [virtual]
```

Obtain a description of the exception.

#### Returns

Description.

Definition at line 35 of file piWithInterrupts.cpp.

References `detailed_message`.

## 6.279.4 Member Data Documentation

### 6.279.4.1 detailed\_message

```
std::string interrupt_exception::detailed_message
```

String with details on the error.

Definition at line 42 of file piWithInterrupts.cpp.

Referenced by what().

The documentation for this class was generated from the following file:

- [inst/examples/OpenMP/piWithInterrupts.cpp](#)

## 6.280 Rcpp::internal::InterruptedException Class Reference

```
#include <Interrupt.h>
```

### 6.280.1 Detailed Description

Definition at line 33 of file Interrupt.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Interrupt.h](#)

## 6.281 Rcpp::sugar::Intersect< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <setdiff.h>
```

### Public Types

- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type STORAGE

## Public Member Functions

- [Intersect](#) (const LHS\_T &lhs, const RHS\_T &rhs)
- [Vector](#)< RTYPE > [get](#) () const

## Private Types

- typedef [RCPP\\_UNORDERED\\_SET](#)< STORAGE > [SET](#)
- typedef SET::const\_iterator [ITERATOR](#)

## Private Attributes

- [SET](#) [intersect](#)

### 6.281.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 100 of file setdiff.h.

### 6.281.2 Member Typedef Documentation

#### 6.281.2.1 ITERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SET::const_iterator Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::ITERATOR
[private]
```

Definition at line 127 of file setdiff.h.

#### 6.281.2.2 SET

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef RCPP\_UNORDERED\_SET<STORAGE> Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T
>::SET [private]
```

Definition at line 126 of file setdiff.h.

### 6.281.2.3 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 102 of file setdiff.h.

## 6.281.3 Constructor & Destructor Documentation

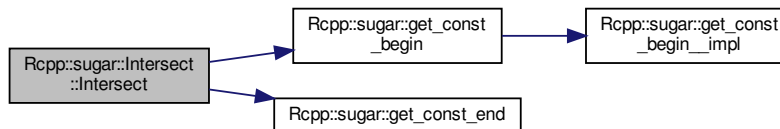
### 6.281.3.1 Intersect()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Intersect (
    const LHS_T & lhs,
    const RHS_T & rhs ) [inline]
```

Definition at line 104 of file setdiff.h.

References [Rcpp::sugar::get\\_const\\_begin\(\)](#), [Rcpp::sugar::get\\_const\\_end\(\)](#), and [Rcpp::sugar::Intersect< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::intersect](#).

Here is the call graph for this function:



### 6.281.4 Member Function Documentation

### 6.281.4.1 get()

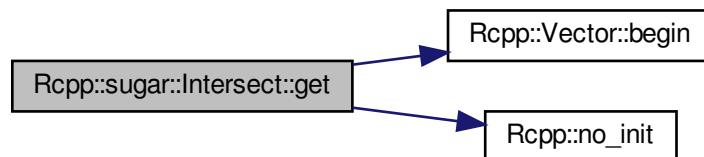
```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Vector<RTYPE> Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get ( ) const [inline]
```

Definition at line 118 of file setdiff.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::sugar::Intersect< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::intersect, and Rcpp::no\_init().

Referenced by Rcpp::intersect().

Here is the call graph for this function:



## 6.281.5 Member Data Documentation

### 6.281.5.1 intersect

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
SET Rcpp::sugar::Intersect< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::intersect [private]
```

Definition at line 128 of file setdiff.h.

Referenced by Rcpp::sugar::Intersect< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::get(), and Rcpp::sugar::Intersect< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::Intersect().

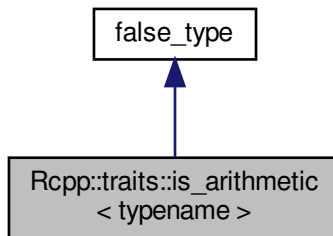
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/setdiff.h

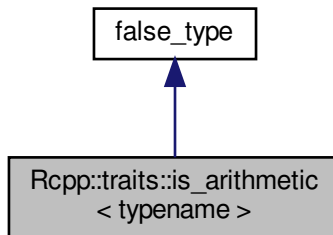
## 6.282 Rcpp::traits::is\_arithmetic< typename > Struct Template Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< typename >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< typename >:



### Additional Inherited Members

#### 6.282.1 Detailed Description

```
template<typename>  
struct Rcpp::traits::is_arithmetic< typename >
```

Definition at line 30 of file `is_arithmetic.h`.

The documentation for this struct was generated from the following file:

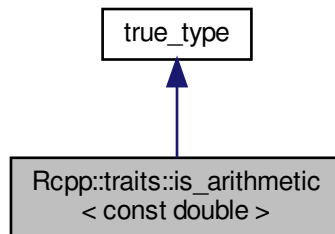
- `inst/include/Rcpp/traits/is_arithmetic.h`



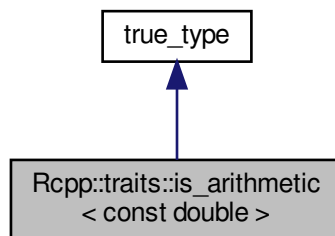
## 6.283 Rcpp::traits::is\_arithmetic< const double > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< const double >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< const double >:



### Additional Inherited Members

#### 6.283.1 Detailed Description

Definition at line 94 of file `is_arithmetic.h`.

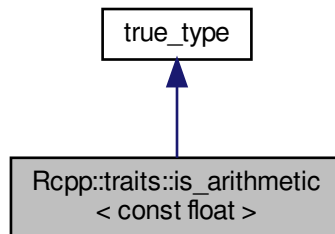
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

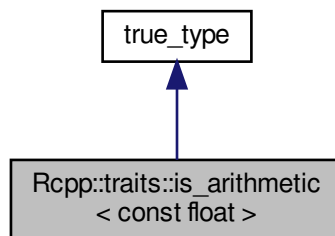
## 6.284 Rcpp::traits::is\_arithmetic< const float > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< const float >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< const float >:



### Additional Inherited Members

#### 6.284.1 Detailed Description

Definition at line 88 of file `is_arithmetic.h`.

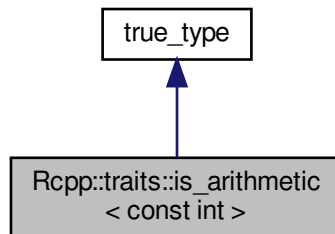
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

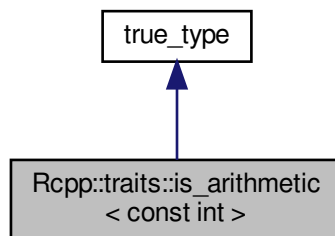
## 6.285 Rcpp::traits::is\_arithmetic< const int > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< const int >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< const int >:



### Additional Inherited Members

#### 6.285.1 Detailed Description

Definition at line 48 of file `is_arithmetic.h`.

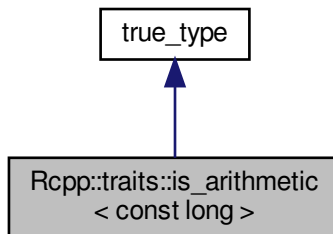
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

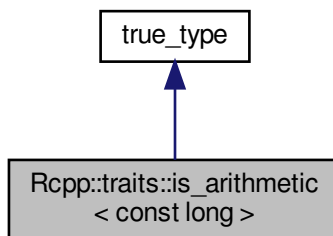
## 6.286 Rcpp::traits::is\_arithmetic< const long > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< const long >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< const long >:



### Additional Inherited Members

#### 6.286.1 Detailed Description

Definition at line 60 of file `is_arithmetic.h`.

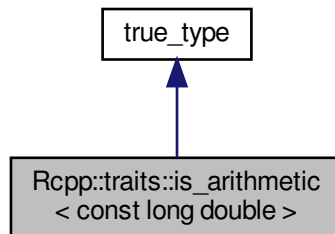
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

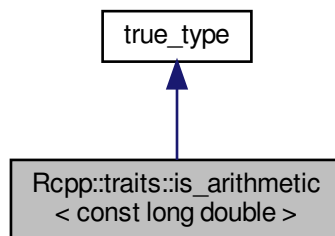
## 6.287 Rcpp::traits::is\_arithmetic< const long double > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< const long double >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< const long double >:



### Additional Inherited Members

#### 6.287.1 Detailed Description

Definition at line 100 of file `is_arithmetic.h`.

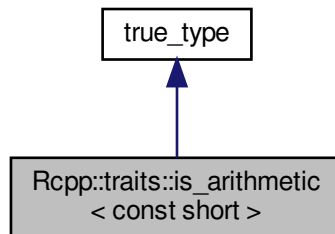
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

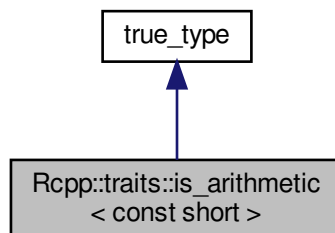
## 6.288 Rcpp::traits::is\_arithmetic< const short > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< const short >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< const short >:



### Additional Inherited Members

#### 6.288.1 Detailed Description

Definition at line 36 of file `is_arithmetic.h`.

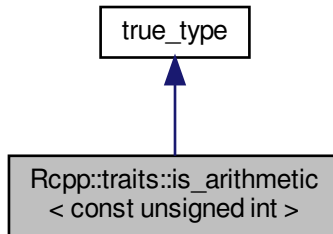
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

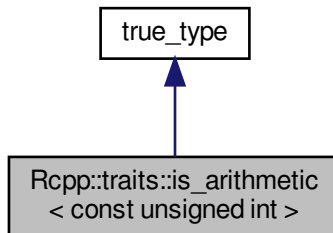
## 6.289 Rcpp::traits::is\_arithmetic< const unsigned int > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< const unsigned int >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< const unsigned int >:



### Additional Inherited Members

#### 6.289.1 Detailed Description

Definition at line 54 of file `is_arithmetic.h`.

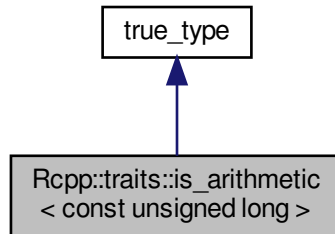
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

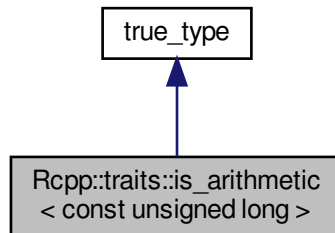
## 6.290 Rcpp::traits::is\_arithmetic< const unsigned long > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< const unsigned long >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< const unsigned long >:



### Additional Inherited Members

#### 6.290.1 Detailed Description

Definition at line 66 of file `is_arithmetic.h`.

The documentation for this struct was generated from the following file:

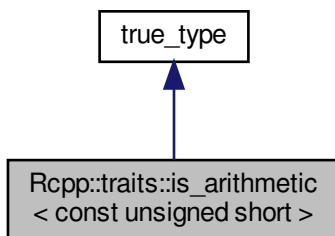
- `inst/include/Rcpp/traits/is_arithmetic.h`



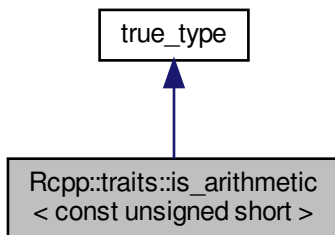
## 6.291 Rcpp::traits::is\_arithmetic< const unsigned short > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< const unsigned short >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< const unsigned short >:



### Additional Inherited Members

#### 6.291.1 Detailed Description

Definition at line 42 of file `is_arithmetic.h`.

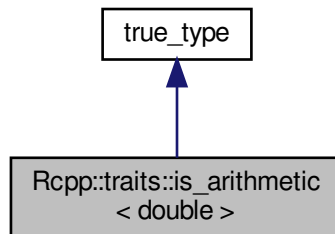
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

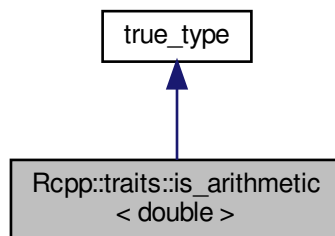
## 6.292 Rcpp::traits::is\_arithmetic< double > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< double >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< double >:



### Additional Inherited Members

#### 6.292.1 Detailed Description

Definition at line 91 of file `is_arithmetic.h`.

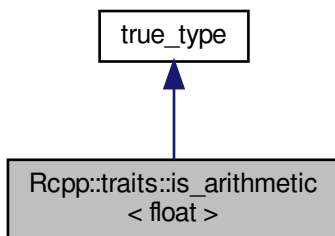
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

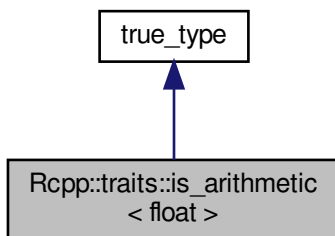
## 6.293 Rcpp::traits::is\_arithmetic< float > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< float >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< float >:



### Additional Inherited Members

#### 6.293.1 Detailed Description

Definition at line 85 of file `is_arithmetic.h`.

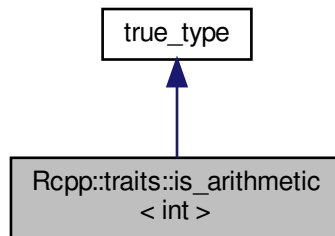
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

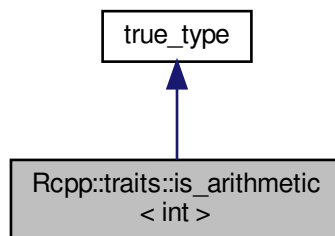
## 6.294 Rcpp::traits::is\_arithmetic< int > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< int >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< int >:



### Additional Inherited Members

#### 6.294.1 Detailed Description

Definition at line 45 of file `is_arithmetic.h`.

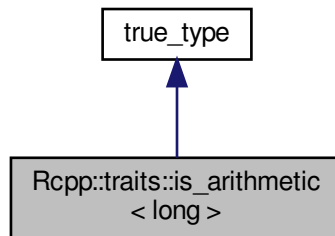
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_arithmetic.h](#)

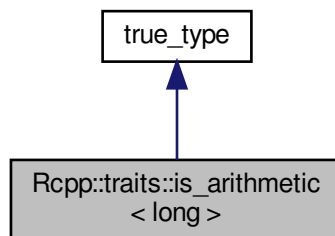
## 6.295 Rcpp::traits::is\_arithmetic< long > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< long >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< long >:



### Additional Inherited Members

#### 6.295.1 Detailed Description

Definition at line 57 of file `is_arithmetic.h`.

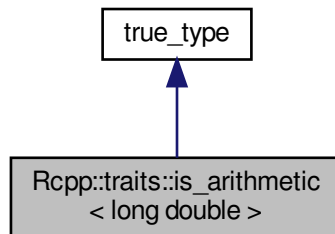
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

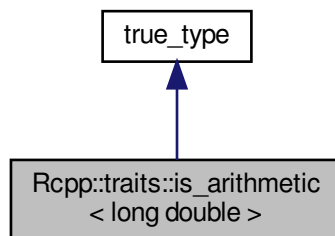
## 6.296 Rcpp::traits::is\_arithmetic< long double > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< long double >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< long double >:



### Additional Inherited Members

#### 6.296.1 Detailed Description

Definition at line 97 of file `is_arithmetic.h`.

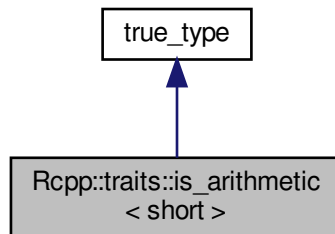
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

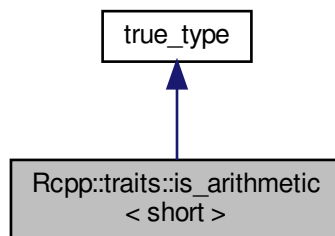
## 6.297 Rcpp::traits::is\_arithmetic< short > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< short >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< short >:



### Additional Inherited Members

#### 6.297.1 Detailed Description

Definition at line 33 of file `is_arithmetic.h`.

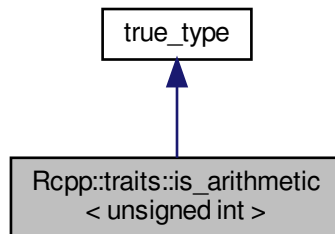
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

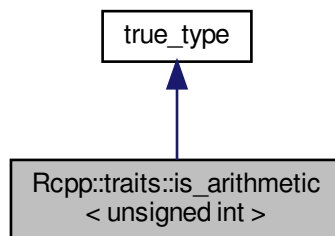
## 6.298 Rcpp::traits::is\_arithmetic< unsigned int > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< unsigned int >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< unsigned int >:



### Additional Inherited Members

#### 6.298.1 Detailed Description

Definition at line 51 of file `is_arithmetic.h`.

The documentation for this struct was generated from the following file:

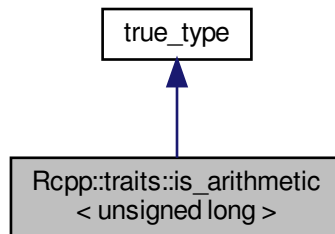
- `inst/include/Rcpp/traits/is_arithmetic.h`



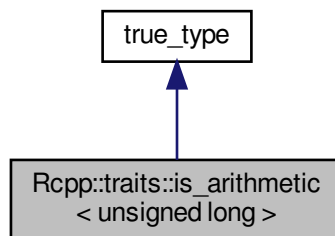
## 6.299 Rcpp::traits::is\_arithmetic< unsigned long > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< unsigned long >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< unsigned long >:



### Additional Inherited Members

#### 6.299.1 Detailed Description

Definition at line 63 of file `is_arithmetic.h`.

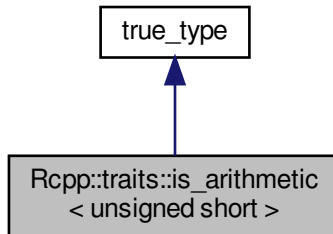
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_arithmetic.h`

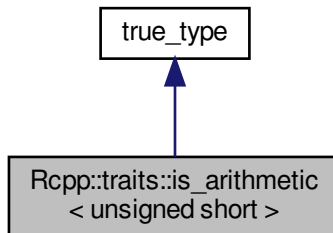
## 6.300 Rcpp::traits::is\_arithmetic< unsigned short > Struct Reference

```
#include <is_arithmetic.h>
```

Inheritance diagram for Rcpp::traits::is\_arithmetic< unsigned short >:



Collaboration diagram for Rcpp::traits::is\_arithmetic< unsigned short >:



### Additional Inherited Members

#### 6.300.1 Detailed Description

Definition at line 39 of file `is_arithmetic.h`.

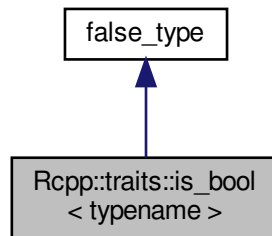
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_arithmetic.h](#)

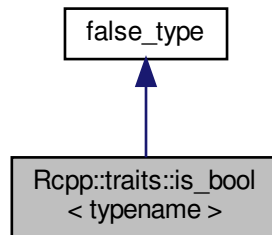
## 6.301 Rcpp::traits::is\_bool< typename > Struct Template Reference

```
#include <is_bool.h>
```

Inheritance diagram for Rcpp::traits::is\_bool< typename >:



Collaboration diagram for Rcpp::traits::is\_bool< typename >:



### Additional Inherited Members

#### 6.301.1 Detailed Description

```
template<typename>  
struct Rcpp::traits::is_bool< typename >
```

Definition at line 7 of file `is_bool.h`.

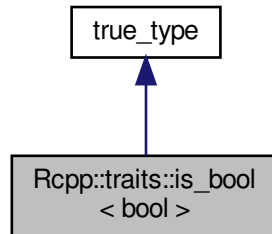
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_bool.h`

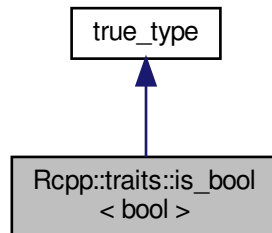
## 6.302 Rcpp::traits::is\_bool< bool > Struct Reference

```
#include <is_bool.h>
```

Inheritance diagram for Rcpp::traits::is\_bool< bool >:



Collaboration diagram for Rcpp::traits::is\_bool< bool >:



### Additional Inherited Members

#### 6.302.1 Detailed Description

Definition at line 11 of file `is_bool.h`.

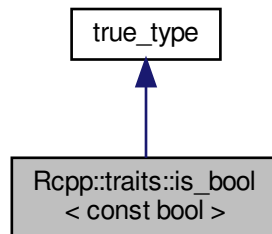
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_bool.h`

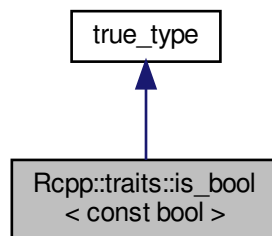
## 6.303 Rcpp::traits::is\_bool< const bool > Struct Reference

```
#include <is_bool.h>
```

Inheritance diagram for Rcpp::traits::is\_bool< const bool >:



Collaboration diagram for Rcpp::traits::is\_bool< const bool >:



### Additional Inherited Members

#### 6.303.1 Detailed Description

Definition at line 15 of file `is_bool.h`.

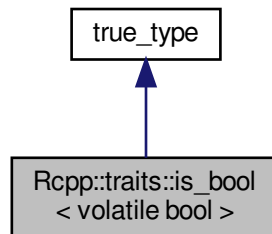
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_bool.h`

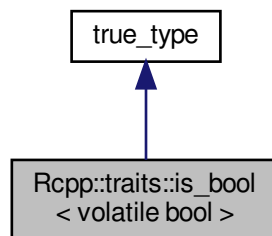
## 6.304 Rcpp::traits::is\_bool< volatile bool > Struct Reference

```
#include <is_bool.h>
```

Inheritance diagram for Rcpp::traits::is\_bool< volatile bool >:



Collaboration diagram for Rcpp::traits::is\_bool< volatile bool >:



### Additional Inherited Members

#### 6.304.1 Detailed Description

Definition at line 19 of file `is_bool.h`.

The documentation for this struct was generated from the following file:

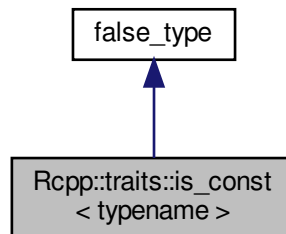
- `inst/include/Rcpp/traits/is_bool.h`

## 6.305 Rcpp::traits::is\_const< typename > Struct Template Reference

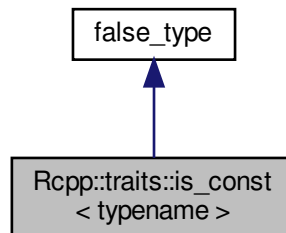
type properties [4.5.3].

```
#include <is_const.h>
```

Inheritance diagram for Rcpp::traits::is\_const< typename >:



Collaboration diagram for Rcpp::traits::is\_const< typename >:



### Additional Inherited Members

#### 6.305.1 Detailed Description

```
template<typename>  
struct Rcpp::traits::is_const< typename >
```

type properties [4.5.3].

Definition at line 30 of file is\_const.h.

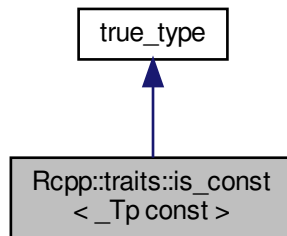
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/is\_const.h

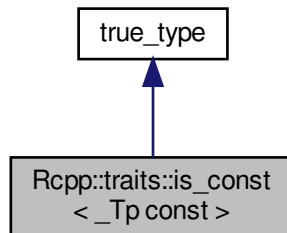
## 6.306 Rcpp::traits::is\_const< \_Tp const > Struct Template Reference

```
#include <is_const.h>
```

Inheritance diagram for Rcpp::traits::is\_const< \_Tp const >:



Collaboration diagram for Rcpp::traits::is\_const< \_Tp const >:



### Additional Inherited Members

#### 6.306.1 Detailed Description

```
template<typename _Tp>  
struct Rcpp::traits::is_const< _Tp const >
```

Definition at line 34 of file `is_const.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_const.h`



## 6.307 Rcpp::traits::is\_convertible< T, U > Class Template Reference

```
#include <is_convertible.h>
```

### Classes

- struct [Big](#)

### Public Types

- typedef bool [value\\_type](#)
- typedef [integral\\_constant](#)< bool, [value](#) > [type](#)

### Static Public Attributes

- static const bool [value](#) = sizeof([Test](#)([MakeT](#)())) == sizeof([Small](#))

### Private Types

- typedef char [Small](#)

### Static Private Member Functions

- static [Small Test](#) (const U &)
- static [Big Test](#) (...)
- static T [MakeT](#) ()

### 6.307.1 Detailed Description

```
template<typename T, typename U>  
class Rcpp::traits::is_convertible< T, U >
```

Definition at line 29 of file `is_convertible.h`.

### 6.307.2 Member Typedef Documentation

### 6.307.2.1 Small

```
template<typename T , typename U >
typedef char Rcpp::traits::is_convertible< T, U >::Small [private]
```

Definition at line 31 of file is\_convertible.h.

### 6.307.2.2 type

```
template<typename T , typename U >
typedef integral_constant<bool,value> Rcpp::traits::is_convertible< T, U >::type
```

Definition at line 41 of file is\_convertible.h.

### 6.307.2.3 value\_type

```
template<typename T , typename U >
typedef bool Rcpp::traits::is_convertible< T, U >::value_type
```

Definition at line 39 of file is\_convertible.h.

## 6.307.3 Member Function Documentation

### 6.307.3.1 MakeT()

```
template<typename T , typename U >
static T Rcpp::traits::is_convertible< T, U >::MakeT ( ) [static], [private]
```

### 6.307.3.2 Test() [1/2]

```
template<typename T , typename U >
static Big Rcpp::traits::is_convertible< T, U >::Test (
    ... ) [static], [private]
```

### 6.307.3.3 Test() [2/2]

```
template<typename T , typename U >
static Small Rcpp::traits::is_convertible< T, U >::Test (
    const U & ) [static], [private]
```

## 6.307.4 Member Data Documentation

### 6.307.4.1 value

```
template<typename T , typename U >
const bool Rcpp::traits::is_convertible< T, U >::value = sizeof(Test (MakeT())) == sizeof(Small)
[static]
```

Definition at line 40 of file is\_convertible.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/is\_convertible.h

## 6.308 tinyformat::detail::is\_convertible< T1, T2 > Struct Template Reference

```
#include <tinyformat.h>
```

### Classes

- struct [fail](#)
- struct [succeed](#)

### Static Public Attributes

- static const bool [value](#)

### Static Private Member Functions

- static [fail tryConvert](#) (...)
- static [succeed tryConvert](#) (const T2 &)
- static const T1 & [makeT1](#) ()

### 6.308.1 Detailed Description

```
template<typename T1, typename T2>  
struct tinyformat::detail::is_convertible< T1, T2 >
```

Definition at line 176 of file tinyformat.h.

### 6.308.2 Member Function Documentation

#### 6.308.2.1 makeT1()

```
template<typename T1 , typename T2 >  
static const T1& tinyformat::detail::is_convertible< T1, T2 >::makeT1 ( ) [static], [private]
```

#### 6.308.2.2 tryConvert() [1/2]

```
template<typename T1 , typename T2 >  
static fail tinyformat::detail::is_convertible< T1, T2 >::tryConvert (   
    ... ) [static], [private]
```

#### 6.308.2.3 tryConvert() [2/2]

```
template<typename T1 , typename T2 >  
static succeed tinyformat::detail::is_convertible< T1, T2 >::tryConvert (   
    const T2 & ) [static], [private]
```

### 6.308.3 Member Data Documentation

### 6.308.3.1 value

```
template<typename T1 , typename T2 >  
const bool tinyformat::detail::is_convertible< T1, T2 >::value [static]
```

**Initial value:**

```
=  
    sizeof(tryConvert(makeT1())) == sizeof(succeed)
```

Definition at line 197 of file tinyformat.h.

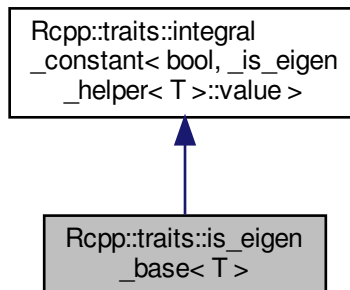
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/utils/tinyformat/tinyformat.h](#)

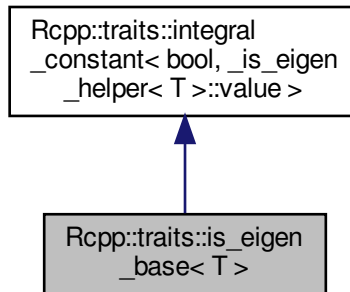
## 6.309 Rcpp::traits::is\_eigen\_base< T > Struct Template Reference

```
#include <is_eigen_base.h>
```

Inheritance diagram for Rcpp::traits::is\_eigen\_base< T >:



Collaboration diagram for `Rcpp::traits::is_eigen_base< T >`:



## Additional Inherited Members

### 6.309.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::is_eigen_base< T >
```

Definition at line 42 of file `is_eigen_base.h`.

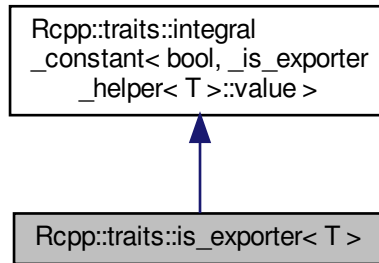
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_eigen\\_base.h](#)

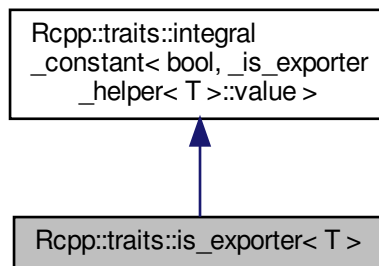
### 6.310 `Rcpp::traits::is_exporter< T >` Struct Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::is\_exporter< T >:



Collaboration diagram for Rcpp::traits::is\_exporter< T >:



## Additional Inherited Members

### 6.310.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::is_exporter< T >
```

Definition at line 115 of file has\_iterator.h.

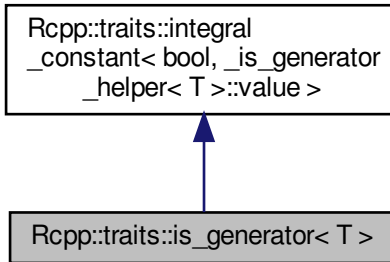
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[has\\_iterator.h](#)

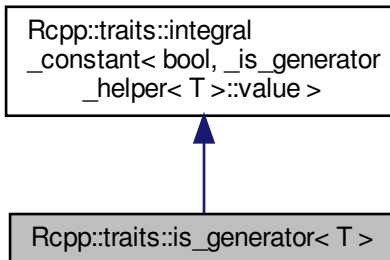
## 6.311 Rcpp::traits::is\_generator< T > Struct Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::is\_generator< T >:



Collaboration diagram for Rcpp::traits::is\_generator< T >:



### Additional Inherited Members

#### 6.311.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::is_generator< T >
```

Definition at line 118 of file `has_iterator.h`.

The documentation for this struct was generated from the following file:

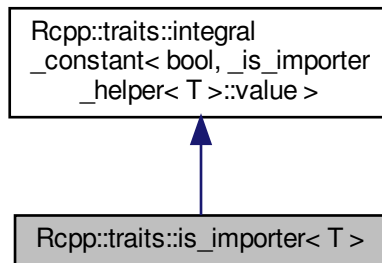
- `inst/include/Rcpp/traits/has_iterator.h`



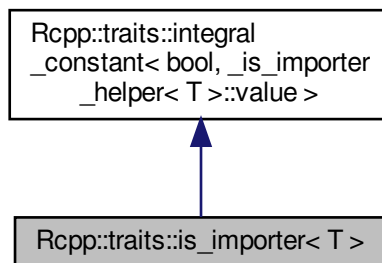
## 6.312 Rcpp::traits::is\_importer< T > Struct Template Reference

```
#include <has_iterator.h>
```

Inheritance diagram for Rcpp::traits::is\_importer< T >:



Collaboration diagram for Rcpp::traits::is\_importer< T >:



### Additional Inherited Members

#### 6.312.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::is_importer< T >
```

uses SFINAE to identify if a type is importable

The test is based on the presence of a typedef `r_import_type` in the class

Definition at line 112 of file `has_iterator.h`.

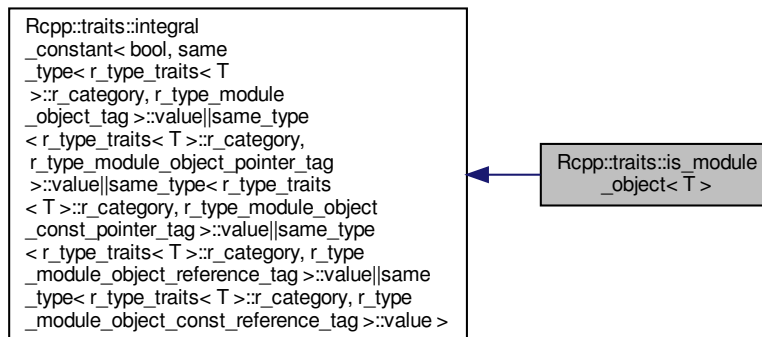
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/has_iterator.h`

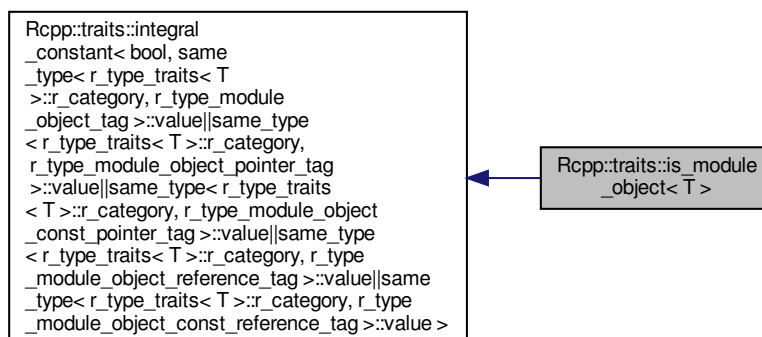
### 6.313 Rcpp::traits::is\_module\_object< T > Struct Template Reference

```
#include <is_module_object.h>
```

Inheritance diagram for `Rcpp::traits::is_module_object< T >`:



Collaboration diagram for `Rcpp::traits::is_module_object< T >`:



## Additional Inherited Members

### 6.313.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::is_module_object< T >
```

Definition at line 29 of file `is_module_object.h`.

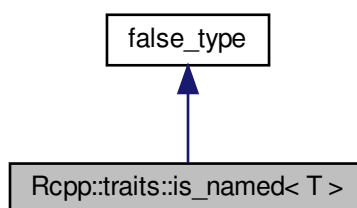
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_module\\_object.h](#)

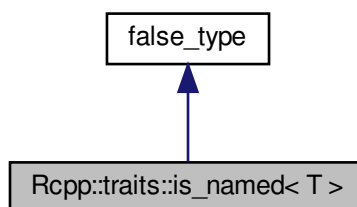
## 6.314 Rcpp::traits::is\_named< T > Struct Template Reference

```
#include <named_object.h>
```

Inheritance diagram for `Rcpp::traits::is_named< T >`:



Collaboration diagram for `Rcpp::traits::is_named< T >`:



## Additional Inherited Members

### 6.314.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::is_named< T >
```

Definition at line 62 of file named\_object.h.

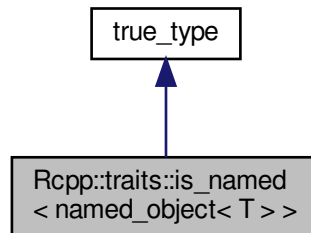
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/named\_object.h

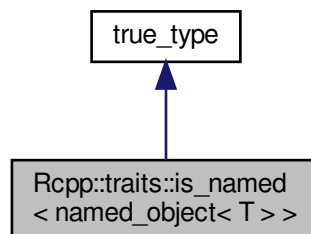
### 6.315 Rcpp::traits::is\_named< named\_object< T > > Struct Template Reference

```
#include <named_object.h>
```

Inheritance diagram for Rcpp::traits::is\_named< named\_object< T > >:



Collaboration diagram for Rcpp::traits::is\_named< named\_object< T > >:



## Additional Inherited Members

### 6.315.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::is_named< named_object< T > >
```

Definition at line 63 of file named\_object.h.

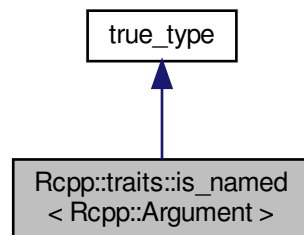
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/named\_object.h

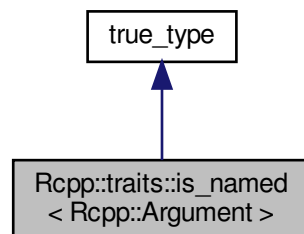
## 6.316 Rcpp::traits::is\_named< Rcpp::Argument > Struct Reference

```
#include <named_object.h>
```

Inheritance diagram for Rcpp::traits::is\_named< Rcpp::Argument >:



Collaboration diagram for Rcpp::traits::is\_named< Rcpp::Argument >:



## Additional Inherited Members

### 6.316.1 Detailed Description

Definition at line 64 of file `named_object.h`.

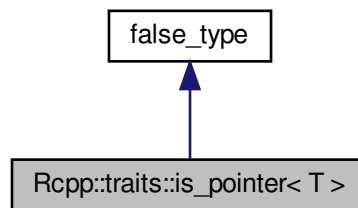
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/named_object.h`

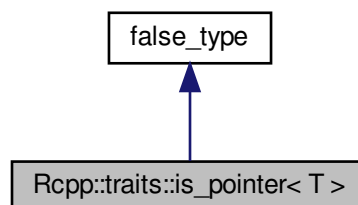
### 6.317 `Rcpp::traits::is_pointer< T >` Struct Template Reference

```
#include <is_pointer.h>
```

Inheritance diagram for `Rcpp::traits::is_pointer< T >`:



Collaboration diagram for `Rcpp::traits::is_pointer< T >`:



## Additional Inherited Members

### 6.317.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::is_pointer< T >
```

Definition at line 29 of file `is_pointer.h`.

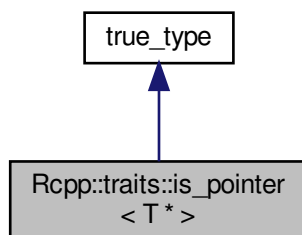
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_pointer.h](#)

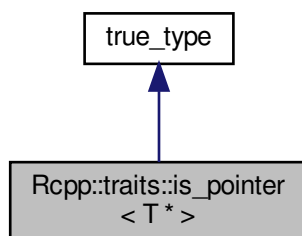
## 6.318 Rcpp::traits::is\_pointer< T \* > Struct Template Reference

```
#include <is_pointer.h>
```

Inheritance diagram for `Rcpp::traits::is_pointer< T * >`:



Collaboration diagram for `Rcpp::traits::is_pointer< T * >`:



## Additional Inherited Members

### 6.318.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::is_pointer< T * >
```

Definition at line 30 of file `is_pointer.h`.

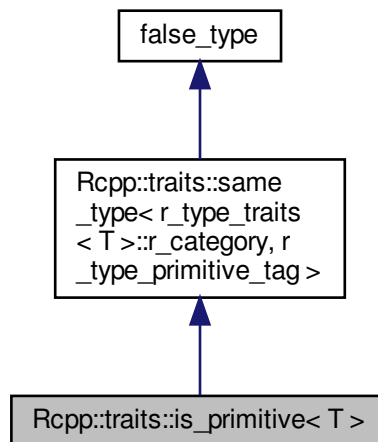
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_pointer.h](#)

### 6.319 Rcpp::traits::is\_primitive< T > Struct Template Reference

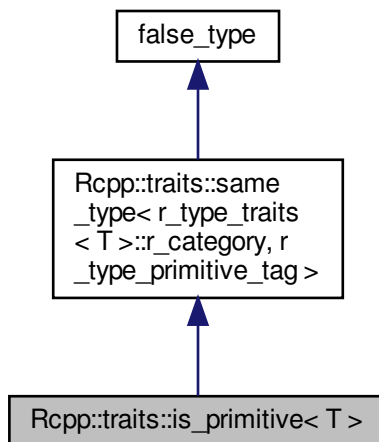
```
#include <is_primitive.h>
```

Inheritance diagram for Rcpp::traits::is\_primitive< T >:





Collaboration diagram for Rcpp::traits::is\_primitive< T >:



## Additional Inherited Members

### 6.319.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::is_primitive< T >
```

Definition at line 29 of file is\_primitive.h.

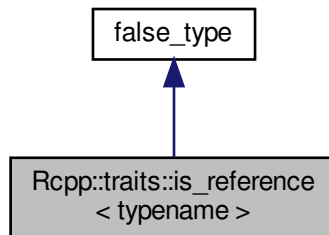
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_primitive.h](#)

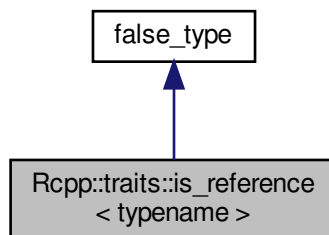
## 6.320 Rcpp::traits::is\_reference< typename > Struct Template Reference

```
#include <is_reference.h>
```

Inheritance diagram for Rcpp::traits::is\_reference< typename >:



Collaboration diagram for Rcpp::traits::is\_reference< typename >:



## Additional Inherited Members

### 6.320.1 Detailed Description

```
template<typename>  
struct Rcpp::traits::is_reference< typename >
```

Definition at line 29 of file `is_reference.h`.

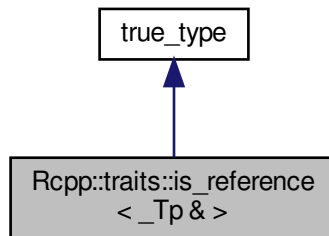
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_reference.h](#)

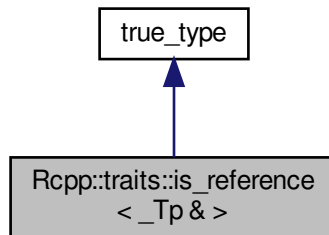
## 6.321 Rcpp::traits::is\_reference<\_Tp &> Struct Template Reference

```
#include <is_reference.h>
```

Inheritance diagram for Rcpp::traits::is\_reference<\_Tp &>:



Collaboration diagram for Rcpp::traits::is\_reference<\_Tp &>:



### Additional Inherited Members

#### 6.321.1 Detailed Description

```
template<typename _Tp>  
struct Rcpp::traits::is_reference<_Tp &>
```

Definition at line 33 of file `is_reference.h`.

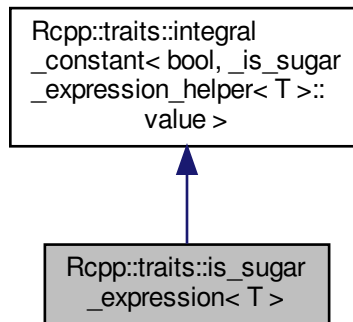
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_reference.h`

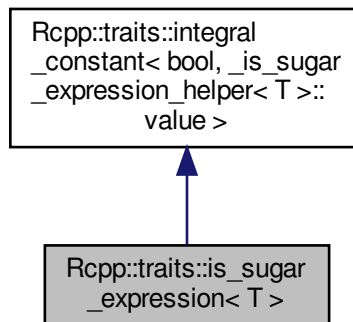
## 6.322 Rcpp::traits::is\_sugar\_expression< T > Struct Template Reference

```
#include <is_sugar_expression.h>
```

Inheritance diagram for Rcpp::traits::is\_sugar\_expression< T >:



Collaboration diagram for Rcpp::traits::is\_sugar\_expression< T >:



### Additional Inherited Members

#### 6.322.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::is_sugar_expression< T >
```

Definition at line 42 of file is\_sugar\_expression.h.

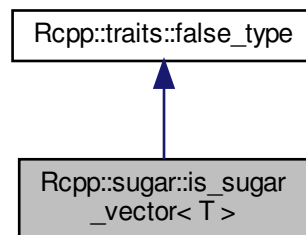
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_sugar\\_expression.h](#)

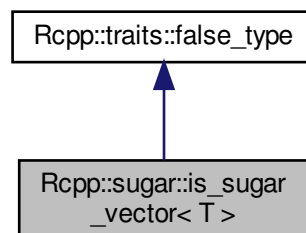
## 6.323 Rcpp::sugar::is\_sugar\_vector< T > Struct Template Reference

```
#include <iterator.h>
```

Inheritance diagram for Rcpp::sugar::is\_sugar\_vector< T >:



Collaboration diagram for Rcpp::sugar::is\_sugar\_vector< T >:



## Additional Inherited Members

### 6.323.1 Detailed Description

```
template<typename T>
struct Rcpp::sugar::is_sugar_vector< T >
```

Definition at line 121 of file iterator.h.

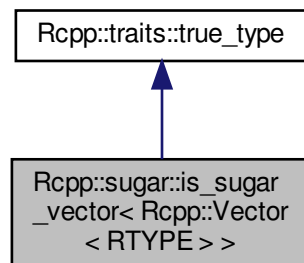
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/tools/iterator.h](#)

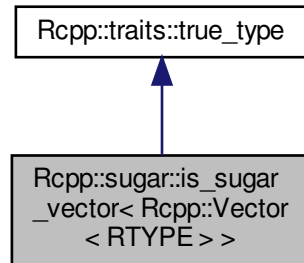
### 6.324 Rcpp::sugar::is\_sugar\_vector< Rcpp::Vector< RTYPE > > Struct Template Reference

```
#include <iterator.h>
```

Inheritance diagram for Rcpp::sugar::is\_sugar\_vector< Rcpp::Vector< RTYPE > >:



Collaboration diagram for Rcpp::sugar::is\_sugar\_vector< Rcpp::Vector< RTYPE > >:



## Additional Inherited Members

### 6.324.1 Detailed Description

```

template<int RTYPE>
struct Rcpp::sugar::is_sugar_vector< Rcpp::Vector< RTYPE > >
  
```

Definition at line 122 of file iterator.h.

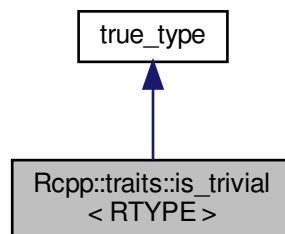
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/tools/iterator.h

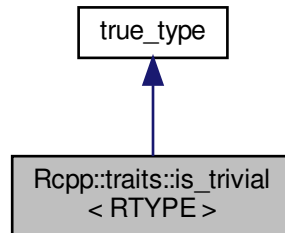
## 6.325 Rcpp::traits::is\_trivial< RTYPE > Struct Template Reference

```
#include <is_trivial.h>
```

Inheritance diagram for Rcpp::traits::is\_trivial< RTYPE >:



Collaboration diagram for `Rcpp::traits::is_trivial< RTYPE >`:



## Additional Inherited Members

### 6.325.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::traits::is_trivial< RTYPE >
```

Definition at line 28 of file `is_trivial.h`.

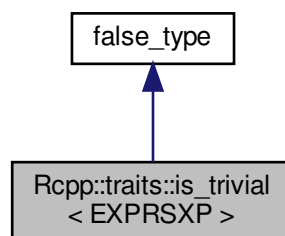
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/is_trivial.h`

### 6.326 `Rcpp::traits::is_trivial< EXPRXP >` Struct Reference

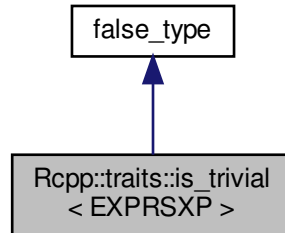
```
#include <is_trivial.h>
```

Inheritance diagram for `Rcpp::traits::is_trivial< EXPRXP >`:





Collaboration diagram for Rcpp::traits::is\_trivial< EXPRXP >:



## Additional Inherited Members

### 6.326.1 Detailed Description

Definition at line 30 of file `is_trivial.h`.

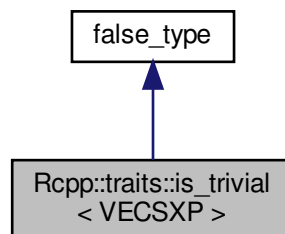
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_trivial.h](#)

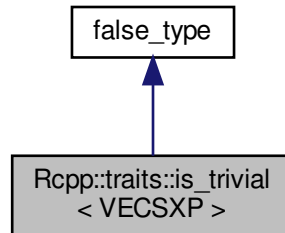
## 6.327 Rcpp::traits::is\_trivial< VECSXP > Struct Reference

```
#include <is_trivial.h>
```

Inheritance diagram for Rcpp::traits::is\_trivial< VECSXP >:



Collaboration diagram for `Rcpp::traits::is_trivial< VECSXP >`:



## Additional Inherited Members

### 6.327.1 Detailed Description

Definition at line 29 of file `is_trivial.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_trivial.h](#)

## 6.328 `tinyformat::detail::is_wchar< T >` Struct Template Reference

```
#include <tinyformat.h>
```

### Public Types

- typedef int [tinyformat\\_wchar\\_is\\_not\\_supported](#)

### 6.328.1 Detailed Description

```
template<typename T>  
struct tinyformat::detail::is_wchar< T >
```

Definition at line 206 of file `tinyformat.h`.

## 6.328.2 Member Typedef Documentation

### 6.328.2.1 `tinyformat_wchar_is_not_supported`

```
template<typename T >
typedef int tinyformat::detail::is_wchar< T >::tinyformat_wchar_is_not_supported
```

Definition at line 206 of file `tinyformat.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/Utils/tinyformat/tinyformat.h`

## 6.329 `tinyformat::detail::is_wchar< const wchar_t * >` Struct Reference

```
#include <tinyformat.h>
```

### 6.329.1 Detailed Description

Definition at line 208 of file `tinyformat.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/Utils/tinyformat/tinyformat.h`

## 6.330 `tinyformat::detail::is_wchar< const wchar_t[n]>` Struct Template Reference

```
#include <tinyformat.h>
```

### 6.330.1 Detailed Description

```
template<int n>
struct tinyformat::detail::is_wchar< const wchar_t[n]>
```

Definition at line 209 of file `tinyformat.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/Utils/tinyformat/tinyformat.h`

### 6.331 `tinyformat::detail::is_wchar< wchar_t * >` Struct Reference

```
#include <tinyformat.h>
```

#### 6.331.1 Detailed Description

Definition at line 207 of file `tinyformat.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/utils/tinyformat/tinyformat.h](#)

### 6.332 `tinyformat::detail::is_wchar< wchar_t[n]>` Struct Template Reference

```
#include <tinyformat.h>
```

#### 6.332.1 Detailed Description

```
template<int n>  
struct tinyformat::detail::is_wchar< wchar_t[n]>
```

Definition at line 210 of file `tinyformat.h`.

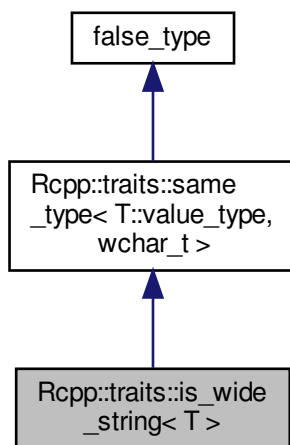
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/utils/tinyformat/tinyformat.h](#)

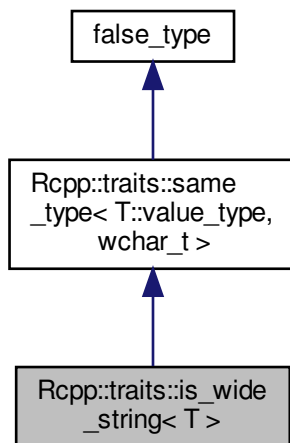
### 6.333 `Rcpp::traits::is_wide_string< T >` Struct Template Reference

```
#include <is_wide_string.h>
```

Inheritance diagram for Rcpp::traits::is\_wide\_string< T >:



Collaboration diagram for Rcpp::traits::is\_wide\_string< T >:



## Additional Inherited Members

### 6.333.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::is_wide_string< T >
```

Definition at line 30 of file `is_wide_string.h`.

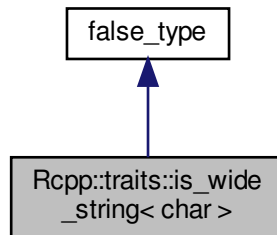
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_wide\\_string.h](#)

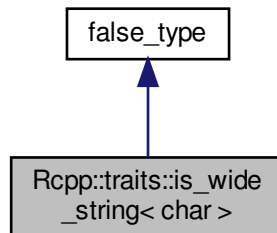
### 6.334 Rcpp::traits::is\_wide\_string< char > Struct Reference

```
#include <is_wide_string.h>
```

Inheritance diagram for Rcpp::traits::is\_wide\_string< char >:



Collaboration diagram for Rcpp::traits::is\_wide\_string< char >:



## Additional Inherited Members

### 6.334.1 Detailed Description

Definition at line 36 of file `is_wide_string.h`.

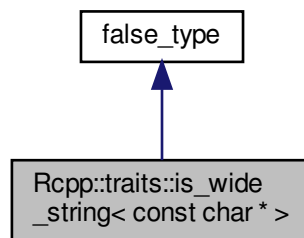
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_wide\\_string.h](#)

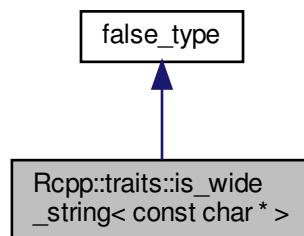
## 6.335 Rcpp::traits::is\_wide\_string< const char \* > Struct Reference

```
#include <is_wide_string.h>
```

Inheritance diagram for `Rcpp::traits::is_wide_string< const char * >`:



Collaboration diagram for `Rcpp::traits::is_wide_string< const char * >`:



## Additional Inherited Members

### 6.335.1 Detailed Description

Definition at line 33 of file `is_wide_string.h`.

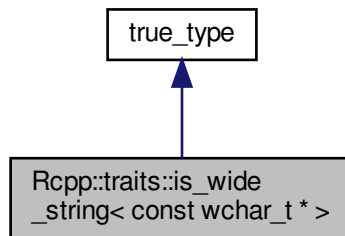
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_wide\\_string.h](#)

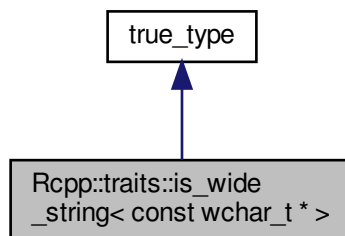
### 6.336 Rcpp::traits::is\_wide\_string< const wchar\_t \* > Struct Reference

```
#include <is_wide_string.h>
```

Inheritance diagram for `Rcpp::traits::is_wide_string< const wchar_t * >`:



Collaboration diagram for `Rcpp::traits::is_wide_string< const wchar_t * >`:





## Additional Inherited Members

### 6.336.1 Detailed Description

Definition at line 32 of file `is_wide_string.h`.

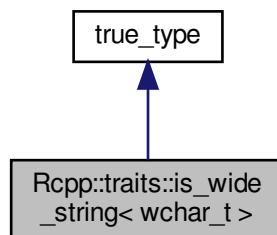
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_wide\\_string.h](#)

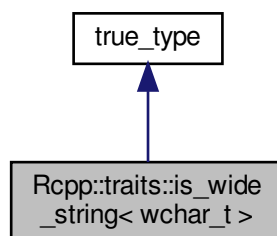
## 6.337 Rcpp::traits::is\_wide\_string< wchar\_t > Struct Reference

```
#include <is_wide_string.h>
```

Inheritance diagram for `Rcpp::traits::is_wide_string< wchar_t >`:



Collaboration diagram for `Rcpp::traits::is_wide_string< wchar_t >`:



## Additional Inherited Members

### 6.337.1 Detailed Description

Definition at line 35 of file `is_wide_string.h`.

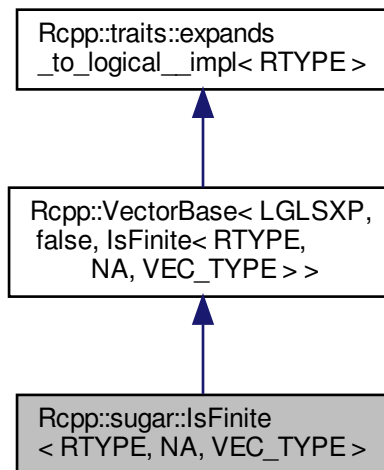
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/is\\_wide\\_string.h](#)

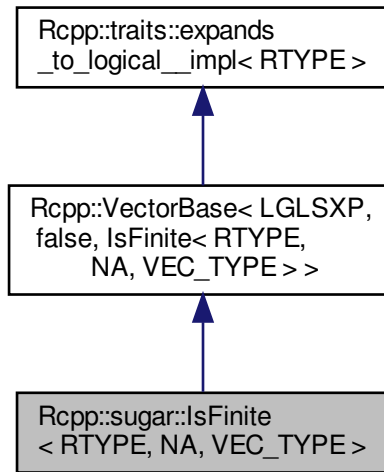
### 6.338 Rcpp::sugar::IsFinite< RTYPE, NA, VEC\_TYPE > Class Template Reference

```
#include <is_finite.h>
```

Inheritance diagram for `Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >`:



Collaboration diagram for Rcpp::sugar::IsFinite< RTYPE, NA, VEC\_TYPE >:



## Public Member Functions

- `IsFinite` (const VEC\_TYPE &obj\_)
- int `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const VEC\_TYPE & `obj`

## Additional Inherited Members

### 6.338.1 Detailed Description

```

template<int RTYPE, bool NA, typename VEC_TYPE>
class Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >
  
```

Definition at line 29 of file `is_finite.h`.

### 6.338.2 Constructor & Destructor Documentation

### 6.338.2.1 IsFinite()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::IsFinite (
    const VEC_TYPE & obj_ ) [inline]
```

Definition at line 32 of file `is_finite.h`.

## 6.338.3 Member Function Documentation

### 6.338.3.1 operator[]()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
int Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 34 of file `is_finite.h`.

References `Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::obj`.

### 6.338.3.2 size()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
R_xlen_t Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::size ( ) const [inline]
```

Definition at line 38 of file `is_finite.h`.

References `Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::obj`.

## 6.338.4 Member Data Documentation

### 6.338.4.1 obj

```
template<int RTYPE, bool NA, typename VEC_TYPE >
const VEC_TYPE& Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::obj [private]
```

Definition at line 41 of file `is_finite.h`.

Referenced by `Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::operator[]()`, and `Rcpp::sugar::IsFinite< RTYPE, NA, VEC_TYPE >::size()`.

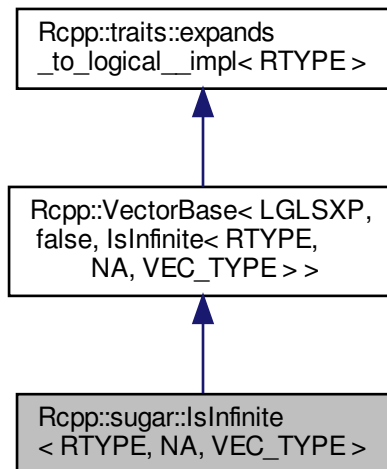
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/is_finite.h`

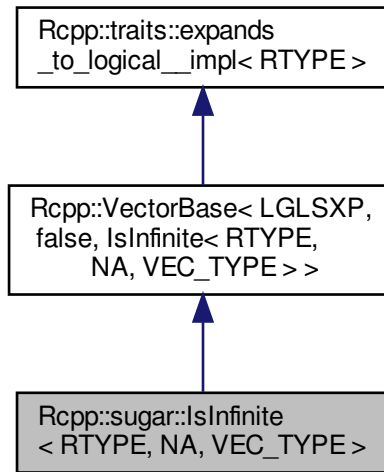
## 6.339 Rcpp::sugar::IsInfinite< RTYPE, NA, VEC\_TYPE > Class Template Reference

```
#include <is_infinite.h>
```

Inheritance diagram for Rcpp::sugar::IsInfinite< RTYPE, NA, VEC\_TYPE >:



Collaboration diagram for `Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >`:



## Public Member Functions

- `IsInfinite` (const VEC\_TYPE &obj\_)
- int `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const VEC\_TYPE & `obj`

## Additional Inherited Members

### 6.339.1 Detailed Description

```

template<int RTYPE, bool NA, typename VEC_TYPE>
class Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >
  
```

Definition at line 29 of file `is_infinite.h`.

### 6.339.2 Constructor & Destructor Documentation

### 6.339.2.1 IsInfinite()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::IsInfinite (
    const VEC_TYPE & obj_ ) [inline]
```

Definition at line 32 of file `is_infinite.h`.

## 6.339.3 Member Function Documentation

### 6.339.3.1 operator[]()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
int Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 34 of file `is_infinite.h`.

References `Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::obj`.

### 6.339.3.2 size()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
R_xlen_t Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::size ( ) const [inline]
```

Definition at line 38 of file `is_infinite.h`.

References `Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::obj`.

## 6.339.4 Member Data Documentation

### 6.339.4.1 obj

```
template<int RTYPE, bool NA, typename VEC_TYPE >
const VEC_TYPE& Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::obj [private]
```

Definition at line 41 of file `is_infinite.h`.

Referenced by `Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::operator[]()`, and `Rcpp::sugar::IsInfinite< RTYPE, NA, VEC_TYPE >::size()`.

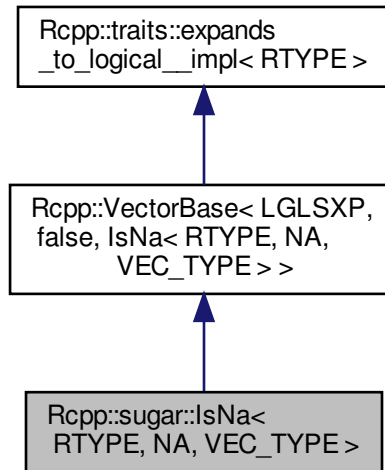
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/is_infinite.h`

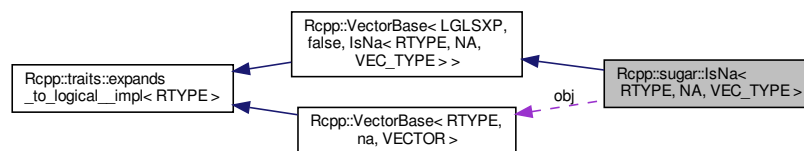
## 6.340 Rcpp::sugar::IsNa< RTYPE, NA, VEC\_TYPE > Class Template Reference

```
#include <is_na.h>
```

Inheritance diagram for Rcpp::sugar::IsNa< RTYPE, NA, VEC\_TYPE >:



Collaboration diagram for Rcpp::sugar::IsNa< RTYPE, NA, VEC\_TYPE >:



### Public Types

- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::VectorBase< RTYPE, NA, VEC_TYPE >` `BASE`



## Public Member Functions

- [IsNa](#) (const [BASE](#) &obj\_)
- int [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [BASE](#) & [obj](#)

### 6.340.1 Detailed Description

```
template<int RTYPE, bool NA, typename VEC_TYPE>
class Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >
```

Definition at line 29 of file [is\\_na.h](#).

### 6.340.2 Member Typedef Documentation

#### 6.340.2.1 BASE

```
template<int RTYPE, bool NA, typename VEC_TYPE >
typedef Rcpp::VectorBase<RTYPE,NA,VEC_TYPE> Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::BASE
```

Definition at line 32 of file [is\\_na.h](#).

#### 6.340.2.2 STORAGE

```
template<int RTYPE, bool NA, typename VEC_TYPE >
typedef traits::storage\_type<RTYPE>::type Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::STORAGE
```

Definition at line 31 of file [is\\_na.h](#).

### 6.340.3 Constructor & Destructor Documentation

### 6.340.3.1 IsNa()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::IsNa (
    const BASE & obj_ ) [inline]
```

Definition at line 34 of file is\_na.h.

## 6.340.4 Member Function Documentation

### 6.340.4.1 operator[]()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
int Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 36 of file is\_na.h.

References Rcpp::sugar::IsNa< RTYPE, NA, VEC\_TYPE >::obj.

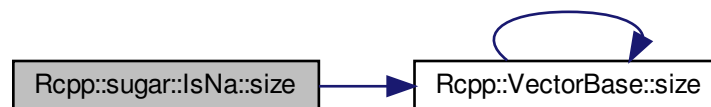
### 6.340.4.2 size()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
R_xlen_t Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::size ( ) const [inline]
```

Definition at line 40 of file is\_na.h.

References Rcpp::sugar::IsNa< RTYPE, NA, VEC\_TYPE >::obj, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



## 6.340.5 Member Data Documentation

### 6.340.5.1 obj

```
template<int RTYPE, bool NA, typename VEC_TYPE >
const BASE& Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::obj [private]
```

Definition at line 43 of file `is_na.h`.

Referenced by `Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::operator[]()`, `Rcpp::sugar::IsNa< RTYPE, NA, VEC_TYPE >::size()`, and `Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::size()`.

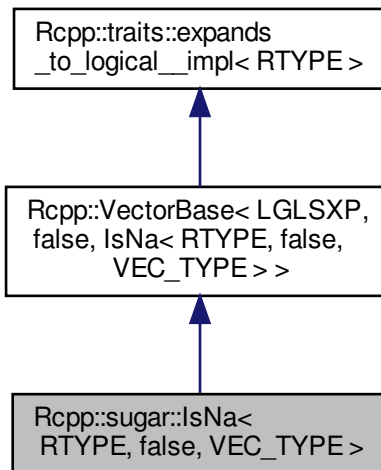
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/is_na.h`

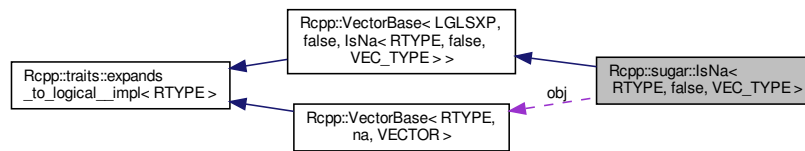
## 6.341 Rcpp::sugar::IsNa< RTYPE, false, VEC\_TYPE > Class Template Reference

```
#include <is_na.h>
```

Inheritance diagram for `Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >`:



Collaboration diagram for `Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >`:



## Public Types

- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::VectorBase< RTYPE, false, VEC_TYPE >` `BASE`

## Public Member Functions

- `IsNa` (const `BASE` &obj\_)
- int `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const `BASE` & `obj`

### 6.341.1 Detailed Description

```

template<int RTYPE, typename VEC_TYPE>
class Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >
  
```

Definition at line 51 of file `is_na.h`.

### 6.341.2 Member Typedef Documentation

#### 6.341.2.1 BASE

```

template<int RTYPE, typename VEC_TYPE >
typedef Rcpp::VectorBase<RTYPE,false,VEC_TYPE> Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::BASE
  
```

Definition at line 54 of file `is_na.h`.

### 6.341.2.2 STORAGE

```
template<int RTYPE, typename VEC_TYPE >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::STORAGE
```

Definition at line 53 of file is\_na.h.

## 6.341.3 Constructor & Destructor Documentation

### 6.341.3.1 IsNa()

```
template<int RTYPE, typename VEC_TYPE >
Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::IsNa (
    const BASE & obj_ ) [inline]
```

Definition at line 56 of file is\_na.h.

## 6.341.4 Member Function Documentation

### 6.341.4.1 operator[]()

```
template<int RTYPE, typename VEC_TYPE >
int Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 58 of file is\_na.h.

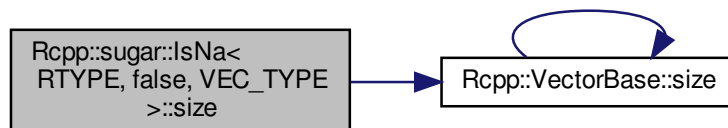
### 6.341.4.2 size()

```
template<int RTYPE, typename VEC_TYPE >
R_xlen_t Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::size ( ) const [inline]
```

Definition at line 62 of file is\_na.h.

References Rcpp::sugar::IsNa< RTYPE, NA, VEC\_TYPE >::obj, and Rcpp::VectorBase< RTYPE, na, VECTOR >←::size().

Here is the call graph for this function:



## 6.341.5 Member Data Documentation

### 6.341.5.1 obj

```
template<int RTYPE, typename VEC_TYPE >
const BASE& Rcpp::sugar::IsNa< RTYPE, false, VEC_TYPE >::obj [private]
```

Definition at line 65 of file `is_na.h`.

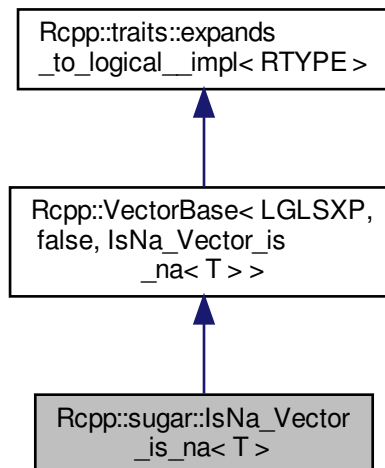
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/is\\_na.h](#)

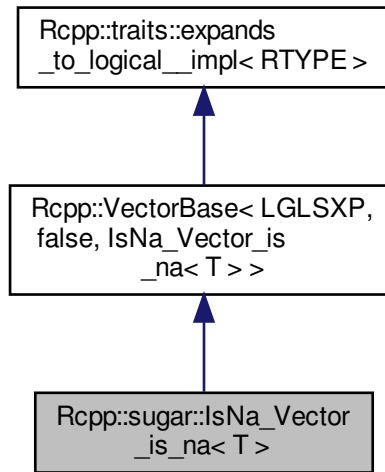
## 6.342 Rcpp::sugar::IsNa\_Vector\_is\_na< T > Class Template Reference

```
#include <is_na.h>
```

Inheritance diagram for `Rcpp::sugar::IsNa_Vector_is_na< T >`:



Collaboration diagram for Rcpp::sugar::IsNa\_Vector\_is\_na< T >:



## Public Member Functions

- [IsNa\\_Vector\\_is\\_na](#) (const T &x)
- int [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const T & [ref](#)

## Additional Inherited Members

### 6.342.1 Detailed Description

```

template<typename T>
class Rcpp::sugar::IsNa_Vector_is_na< T >
  
```

Definition at line 70 of file `is_na.h`.

### 6.342.2 Constructor & Destructor Documentation

### 6.342.2.1 IsNa\_Vector\_is\_na()

```
template<typename T >
Rcpp::sugar::IsNa_Vector_is_na< T >::IsNa_Vector_is_na (
    const T & x ) [inline]
```

Definition at line 72 of file `is_na.h`.

## 6.342.3 Member Function Documentation

### 6.342.3.1 operator[]()

```
template<typename T >
int Rcpp::sugar::IsNa_Vector_is_na< T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 74 of file `is_na.h`.

References `Rcpp::sugar::IsNa_Vector_is_na< T >::ref`.

### 6.342.3.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::IsNa_Vector_is_na< T >::size ( ) const [inline]
```

Definition at line 82 of file `is_na.h`.

References `Rcpp::sugar::IsNa_Vector_is_na< T >::ref`.

## 6.342.4 Member Data Documentation

### 6.342.4.1 ref

```
template<typename T >
const T& Rcpp::sugar::IsNa_Vector_is_na< T >::ref [private]
```

Definition at line 85 of file `is_na.h`.

Referenced by `Rcpp::sugar::IsNa_Vector_is_na< T >::operator[]()`, and `Rcpp::sugar::IsNa_Vector_is_na< T >::size()`.

The documentation for this class was generated from the following file:

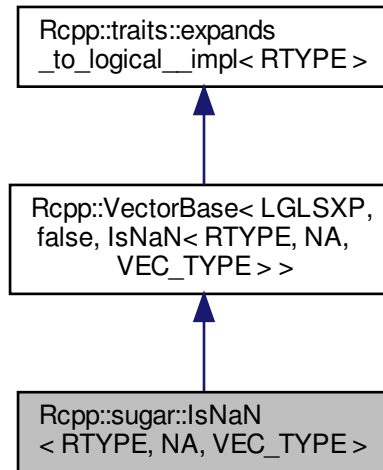
- `inst/include/Rcpp/sugar/functions/is_na.h`



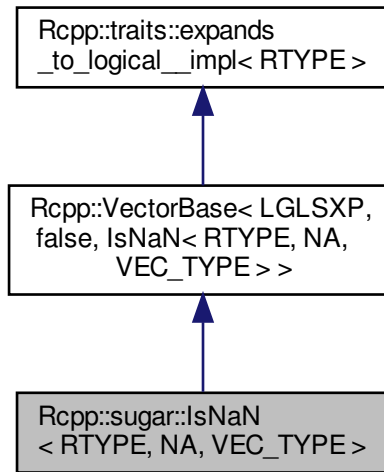
## 6.343 Rcpp::sugar::IsNaN< RTYPE, NA, VEC\_TYPE > Class Template Reference

```
#include <is_nan.h>
```

Inheritance diagram for Rcpp::sugar::IsNaN< RTYPE, NA, VEC\_TYPE >:



Collaboration diagram for `Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >`:



## Public Member Functions

- `IsNaN` (const `VEC_TYPE` &obj\_)
- int `operator[]` (`R_xlen_t` i) const
- `R_xlen_t` `size` () const

## Private Attributes

- const `VEC_TYPE` & `obj`

## Additional Inherited Members

### 6.343.1 Detailed Description

```

template<int RTYPE, bool NA, typename VEC_TYPE>
class Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >
  
```

Definition at line 29 of file `is_nan.h`.

### 6.343.2 Constructor & Destructor Documentation

### 6.343.2.1 IsNaN()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::IsNaN (
    const VEC_TYPE & obj_ ) [inline]
```

Definition at line 32 of file `is_nan.h`.

## 6.343.3 Member Function Documentation

### 6.343.3.1 operator[]()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
int Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 34 of file `is_nan.h`.

References `Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::obj`.

### 6.343.3.2 size()

```
template<int RTYPE, bool NA, typename VEC_TYPE >
R_xlen_t Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::size ( ) const [inline]
```

Definition at line 38 of file `is_nan.h`.

References `Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::obj`.

## 6.343.4 Member Data Documentation

### 6.343.4.1 obj

```
template<int RTYPE, bool NA, typename VEC_TYPE >
const VEC_TYPE& Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::obj [private]
```

Definition at line 41 of file `is_nan.h`.

Referenced by `Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::operator[]()`, and `Rcpp::sugar::IsNaN< RTYPE, NA, VEC_TYPE >::size()`.

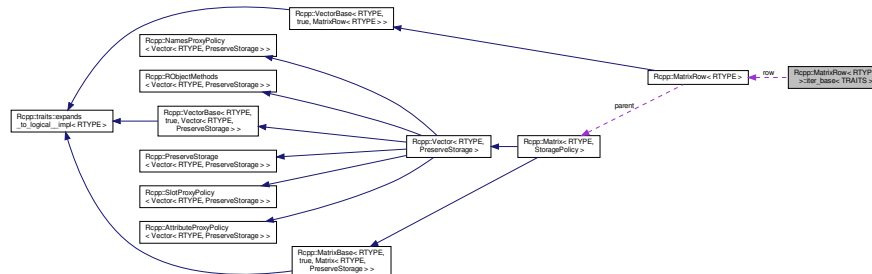
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/is_nan.h`

## 6.344 Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS > Class Template Reference

```
#include <MatrixRow.h>
```

Collaboration diagram for Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >:



### Public Types

- typedef TRAITS::vector\_iterator [vector\\_iterator](#)
- typedef TRAITS::difference\_type [difference\\_type](#)
- typedef TRAITS::value\_type [value\\_type](#)
- typedef TRAITS::reference [reference](#)
- typedef TRAITS::pointer [pointer](#)
- typedef std::random\_access\_iterator\_tag [iterator\\_category](#)

### Public Member Functions

- [iter\\_base](#) (const [iter\\_base](#) &other)
- [iter\\_base](#) ([MatrixRow](#) &row\_, int index\_)
- [iter\\_base](#) & operator++ ()
- [iter\\_base](#) operator++ (int)
- [iter\\_base](#) & operator-- ()
- [iter\\_base](#) operator-- (int)
- [iter\\_base](#) operator+ ([difference\\_type](#) n) const
- [iter\\_base](#) operator- ([difference\\_type](#) n) const
- [difference\\_type](#) operator- (const [iter\\_base](#) &other) const
- [iter\\_base](#) & operator+= ([difference\\_type](#) n)
- [iter\\_base](#) & operator-= ([difference\\_type](#) n)
- [reference](#) operator\* ()
- [pointer](#) operator-> ()
- bool operator== (const [iter\\_base](#) &other)
- bool operator!= (const [iter\\_base](#) &other)
- bool operator< (const [iter\\_base](#) &other)
- bool operator> (const [iter\\_base](#) &other)
- bool operator<= (const [iter\\_base](#) &other)
- bool operator>= (const [iter\\_base](#) &other)
- [reference](#) operator[] (int i) const
- [difference\\_type](#) operator- (const [iter\\_base](#) &other)

## Private Attributes

- [MatrixRow](#) & [row](#)
- int [index](#)

### 6.344.1 Detailed Description

```
template<int RTYPE>
template<typename TRAITS>
class Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >
```

Definition at line 62 of file MatrixRow.h.

### 6.344.2 Member Typedef Documentation

#### 6.344.2.1 difference\_type

```
template<int RTYPE>
template<typename TRAITS >
typedef TRAITS::difference_type Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::difference_type
```

Definition at line 66 of file MatrixRow.h.

#### 6.344.2.2 iterator\_category

```
template<int RTYPE>
template<typename TRAITS >
typedef std::random_access_iterator_tag Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::iterator_category
```

Definition at line 71 of file MatrixRow.h.

#### 6.344.2.3 pointer

```
template<int RTYPE>
template<typename TRAITS >
typedef TRAITS::pointer Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::pointer
```

Definition at line 69 of file MatrixRow.h.

#### 6.344.2.4 reference

```
template<int RTYPE>
template<typename TRAITS >
typedef TRAITS::reference Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::reference
```

Definition at line 68 of file MatrixRow.h.

#### 6.344.2.5 value\_type

```
template<int RTYPE>
template<typename TRAITS >
typedef TRAITS::value_type Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::value_type
```

Definition at line 67 of file MatrixRow.h.

#### 6.344.2.6 vector\_iterator

```
template<int RTYPE>
template<typename TRAITS >
typedef TRAITS::vector_iterator Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::vector_iterator
```

Definition at line 64 of file MatrixRow.h.

### 6.344.3 Constructor & Destructor Documentation

#### 6.344.3.1 iter\_base() [1/2]

```
template<int RTYPE>
template<typename TRAITS >
Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::iter_base (
    const iter_base< TRAITS > & other ) [inline]
```

Definition at line 73 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator+(), and Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator-().

### 6.344.3.2 iter\_base() [2/2]

```
template<int RTYPE>
template<typename TRAITS >
Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::iter_base (
    MatrixRow & row_,
    int index_ ) [inline]
```

Definition at line 74 of file MatrixRow.h.

## 6.344.4 Member Function Documentation

### 6.344.4.1 operator"!=(())

```
template<int RTYPE>
template<typename TRAITS >
bool Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator!= (
    const iter_base< TRAITS > & other ) [inline]
```

Definition at line 111 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index.

### 6.344.4.2 operator\*()

```
template<int RTYPE>
template<typename TRAITS >
reference Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator* ( ) [inline]
```

Definition at line 103 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index, and Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::row.

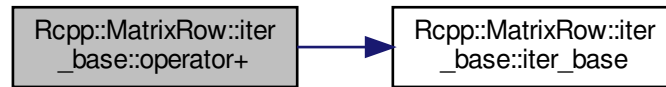
### 6.344.4.3 operator+()

```
template<int RTYPE>
template<typename TRAITS >
iter_base Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator+ (
    difference_type n ) const [inline]
```

Definition at line 96 of file MatrixRow.h.

References `Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index`, `Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::iter_base()`, and `Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::row`.

Here is the call graph for this function:



### 6.344.4.4 operator++() [1/2]

```
template<int RTYPE>
template<typename TRAITS >
iter_base& Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator++ ( ) [inline]
```

Definition at line 76 of file MatrixRow.h.

References `Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index`.

### 6.344.4.5 operator++() [2/2]

```
template<int RTYPE>
template<typename TRAITS >
iter_base Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator++ (
    int ) [inline]
```

Definition at line 80 of file MatrixRow.h.

References `Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index`.



#### 6.344.4.6 operator+=( )

```
template<int RTYPE>
template<typename TRAITS >
iter_base& Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator+=(
    difference_type n ) [inline]
```

Definition at line 100 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index.

#### 6.344.4.7 operator-( ) [1/3]

```
template<int RTYPE>
template<typename TRAITS >
difference_type Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator- (
    const iter_base< TRAITS > & other ) [inline]
```

Definition at line 121 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index.

#### 6.344.4.8 operator-( ) [2/3]

```
template<int RTYPE>
template<typename TRAITS >
difference_type Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator- (
    const iter_base< TRAITS > & other ) const [inline]
```

Definition at line 98 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index.

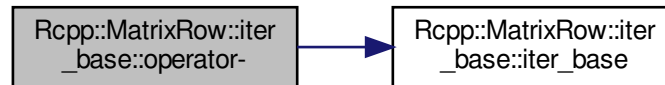
**6.344.4.9 operator-() [3/3]**

```
template<int RTYPE>
template<typename TRAITS >
iter_base Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator- (
    difference_type n ) const [inline]
```

Definition at line 97 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index, Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::iter\_base(), and Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::row.

Here is the call graph for this function:

**6.344.4.10 operator--() [1/2]**

```
template<int RTYPE>
template<typename TRAITS >
iter_base& Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator-- ( ) [inline]
```

Definition at line 86 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index.

**6.344.4.11 operator--() [2/2]**

```
template<int RTYPE>
template<typename TRAITS >
iter_base Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator-- (
    int ) [inline]
```

Definition at line 90 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index.

#### 6.344.4.12 operator-=( )

```
template<int RTYPE>
template<typename TRAITS >
iter_base& Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator-= (
    difference_type n ) [inline]
```

Definition at line 101 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index.

#### 6.344.4.13 operator->( )

```
template<int RTYPE>
template<typename TRAITS >
pointer Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator-> ( ) [inline]
```

Definition at line 106 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index, and Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::row.

#### 6.344.4.14 operator<( )

```
template<int RTYPE>
template<typename TRAITS >
bool Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator< (
    const iter_base< TRAITS > & other ) [inline]
```

Definition at line 112 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index.

#### 6.344.4.15 operator<=( )

```
template<int RTYPE>
template<typename TRAITS >
bool Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator<= (
    const iter_base< TRAITS > & other ) [inline]
```

Definition at line 114 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index.

#### 6.344.4.16 operator==( )

```
template<int RTYPE>
template<typename TRAITS >
bool Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator==(
    const iter_base< TRAITS > & other ) [inline]
```

Definition at line 110 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index.

#### 6.344.4.17 operator>( )

```
template<int RTYPE>
template<typename TRAITS >
bool Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator> (
    const iter_base< TRAITS > & other ) [inline]
```

Definition at line 113 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index.

#### 6.344.4.18 operator>=( )

```
template<int RTYPE>
template<typename TRAITS >
bool Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator>= (
    const iter_base< TRAITS > & other ) [inline]
```

Definition at line 115 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index.

#### 6.344.4.19 operator[]( )

```
template<int RTYPE>
template<typename TRAITS >
reference Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::operator[] (
    int i ) const [inline]
```

Definition at line 117 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::index, and Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::row.

## 6.344.5 Member Data Documentation

### 6.344.5.1 index

```
template<int RTYPE>
template<typename TRAITS >
int Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::index [private]
```

Definition at line 127 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator!==( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator\*( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator+( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator++( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator+=( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator-( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator--( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator-=( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator->( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator<( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator<=( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator==( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator>( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator>=( ), and Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator[ ]( ).

### 6.344.5.2 row

```
template<int RTYPE>
template<typename TRAITS >
MatrixRow& Rcpp::MatrixRow< RTYPE >::iter_base< TRAITS >::row [private]
```

Definition at line 126 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator\*( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator+( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator-( ), Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator->( ), and Rcpp::MatrixRow< RTYPE >::iter\_base< TRAITS >::operator[ ]( ).

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixRow.h

## 6.345 Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS > Class Template Reference

```
#include <VectorBase.h>
```

## Public Types

- typedef TRAITS::reference [reference](#)
- typedef TRAITS::pointer [pointer](#)
- typedef TRAITS::difference\_type [difference\\_type](#)
- typedef TRAITS::value\_type [value\\_type](#)
- typedef TRAITS::iterator\_category [iterator\\_category](#)

## Public Member Functions

- [iter\\_base](#) (const [VectorBase](#) &object\_, R\_xlen\_t index\_)
- [iter\\_base](#) & [operator++](#) ()
- [iter\\_base](#) [operator++](#) (int)
- [iter\\_base](#) & [operator--](#) ()
- [iter\\_base](#) [operator--](#) (int)
- [iter\\_base](#) [operator+](#) ([difference\\_type](#) n) const
- [iter\\_base](#) [operator-](#) ([difference\\_type](#) n) const
- [iter\\_base](#) & [operator+=](#) ([difference\\_type](#) n)
- [iter\\_base](#) & [operator-=](#) ([difference\\_type](#) n)
- [reference](#) [operator\[\]](#) (R\_xlen\_t i)
- [reference](#) [operator\\*](#) ()
- [pointer](#) [operator->](#) ()
- bool [operator==](#) (const [iter\\_base](#) &y) const
- bool [operator!=](#) (const [iter\\_base](#) &y) const
- bool [operator<](#) (const [iter\\_base](#) &other) const
- bool [operator>](#) (const [iter\\_base](#) &other) const
- bool [operator<=](#) (const [iter\\_base](#) &other) const
- bool [operator>=](#) (const [iter\\_base](#) &other) const
- [difference\\_type](#) [operator-](#) (const [iter\\_base](#) &other) const

## Private Attributes

- const VECTOR & [object](#)
- R\_xlen\_t [index](#)

### 6.345.1 Detailed Description

```
template<int RTYPE, bool na, typename VECTOR>
template<typename TRAITS>
class Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >
```

Definition at line 75 of file VectorBase.h.

### 6.345.2 Member Typedef Documentation

### 6.345.2.1 difference\_type

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
typedef TRAITS::difference_type Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >↔
::difference_type
```

Definition at line 79 of file VectorBase.h.

### 6.345.2.2 iterator\_category

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
typedef TRAITS::iterator_category Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >↔
::iterator_category
```

Definition at line 81 of file VectorBase.h.

### 6.345.2.3 pointer

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
typedef TRAITS::pointer Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::pointer
```

Definition at line 78 of file VectorBase.h.

### 6.345.2.4 reference

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
typedef TRAITS::reference Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::reference
```

Definition at line 77 of file VectorBase.h.

### 6.345.2.5 value\_type

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
typedef TRAITS::value_type Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::value_type
```

Definition at line 80 of file VectorBase.h.

## 6.345.3 Constructor & Destructor Documentation

### 6.345.3.1 iter\_base()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::iter_base (
    const VectorBase & object_,
    R_xlen_t index_ ) [inline]
```

Definition at line 83 of file VectorBase.h.

## 6.345.4 Member Function Documentation

### 6.345.4.1 operator"!="()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator!= (
    const iter_base< TRAITS > & y ) const [inline]
```

Definition at line 135 of file VectorBase.h.

References [Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\\_base< TRAITS >::index](#).

### 6.345.4.2 operator\*()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
reference Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator* ( ) [inline]
```

Definition at line 125 of file VectorBase.h.

References [Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\\_base< TRAITS >::index](#).



### 6.345.4.3 operator+()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator+ (
    difference_type n ) const [inline]
```

Definition at line 105 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

### 6.345.4.4 operator++() [1/2]

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base& Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator++ ( ) [inline]
```

Definition at line 85 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

### 6.345.4.5 operator++() [2/2]

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator++ (
    int ) [inline]
```

Definition at line 89 of file VectorBase.h.

### 6.345.4.6 operator+=()

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base& Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator+= (
    difference_type n ) [inline]
```

Definition at line 112 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

#### 6.345.4.7 operator-() [1/2]

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
difference_type Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator- (
    const iter_base< TRAITS > & other ) const [inline]
```

Definition at line 151 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

#### 6.345.4.8 operator-() [2/2]

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator- (
    difference_type n ) const [inline]
```

Definition at line 108 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

#### 6.345.4.9 operator--() [1/2]

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base& Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator-- ( ) [inline]
```

Definition at line 95 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

#### 6.345.4.10 operator--() [2/2]

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator-- (
    int ) [inline]
```

Definition at line 99 of file VectorBase.h.

#### 6.345.4.11 operator-=( )

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
iter_base& Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator-= (
    difference_type n ) [inline]
```

Definition at line 116 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

#### 6.345.4.12 operator->( )

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
pointer Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator-> ( ) [inline]
```

Definition at line 128 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

#### 6.345.4.13 operator<( )

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator< (
    const iter_base< TRAITS > & other ) const [inline]
```

Definition at line 138 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

#### 6.345.4.14 operator<=( )

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator<= (
    const iter_base< TRAITS > & other ) const [inline]
```

Definition at line 144 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

#### 6.345.4.15 operator==( )

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator==(
    const iter_base< TRAITS > & y ) const [inline]
```

Definition at line 132 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

#### 6.345.4.16 operator>( )

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator> (
    const iter_base< TRAITS > & other ) const [inline]
```

Definition at line 141 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

#### 6.345.4.17 operator>=( )

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
bool Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator>= (
    const iter_base< TRAITS > & other ) const [inline]
```

Definition at line 147 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

#### 6.345.4.18 operator[]( )

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
reference Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::operator[] (
    R_xlen_t i ) [inline]
```

Definition at line 121 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::index.

## 6.345.5 Member Data Documentation

### 6.345.5.1 index

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
R_xlen_t Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::index [private]
```

Definition at line 158 of file VectorBase.h.

Referenced by Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator!(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator\*(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator+(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator++(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator+=(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator-(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator--(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator-=(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator->(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator<(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator<=(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator==(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator>(), Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator>=(), and Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_base< TRAITS >::operator[]().

### 6.345.5.2 object

```
template<int RTYPE, bool na, typename VECTOR >
template<typename TRAITS >
const VECTOR& Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_base< TRAITS >::object [private]
```

Definition at line 157 of file VectorBase.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/vector/VectorBase.h](#)

## 6.346 Rcpp::MatrixRow< RTYPE >::iter\_traits Struct Reference

```
#include <MatrixRow.h>
```

### Public Types

- typedef [traits::r\\_vector\\_iterator](#)< RTYPE >::type [vector\\_iterator](#)
- typedef int [difference\\_type](#)
- typedef [traits::r\\_vector\\_proxy](#)< RTYPE >::type [value\\_type](#)
- typedef [value\\_type](#) [reference](#)
- typedef std::iterator\_traits< [vector\\_iterator](#) >::pointer [pointer](#)

### 6.346.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::MatrixRow< RTYPE >::iter_traits
```

Definition at line 41 of file MatrixRow.h.

### 6.346.2 Member Typedef Documentation

#### 6.346.2.1 difference\_type

```
template<int RTYPE>
typedef int Rcpp::MatrixRow< RTYPE >::iter_traits::difference_type
```

Definition at line 45 of file MatrixRow.h.

#### 6.346.2.2 pointer

```
template<int RTYPE>
typedef std::iterator_traits<vector_iterator>::pointer Rcpp::MatrixRow< RTYPE >::iter_traits::pointer
```

Definition at line 48 of file MatrixRow.h.

#### 6.346.2.3 reference

```
template<int RTYPE>
typedef value_type Rcpp::MatrixRow< RTYPE >::iter_traits::reference
```

Definition at line 47 of file MatrixRow.h.

#### 6.346.2.4 value\_type

```
template<int RTYPE>
typedef traits::r_vector_proxy<RTYPE>::type Rcpp::MatrixRow< RTYPE >::iter_traits::value_type
```

Definition at line 46 of file MatrixRow.h.

### 6.346.2.5 vector\_iterator

```
template<int RTYPE>
typedef traits::r_vector_iterator<RTYPE>::type Rcpp::MatrixRow< RTYPE >::iter_traits::vector_iterator
```

Definition at line 43 of file MatrixRow.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/MatrixRow.h](#)

## 6.347 Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\_traits Struct Reference

```
#include <VectorBase.h>
```

### Public Types

- typedef [stored\\_type](#) & [reference](#)
- typedef [stored\\_type](#) \* [pointer](#)
- typedef R\_xlen\_t [difference\\_type](#)
- typedef [stored\\_type](#) [value\\_type](#)
- typedef std::random\_access\_iterator\_tag [iterator\\_category](#)

### 6.347.1 Detailed Description

```
template<int RTYPE, bool na, typename VECTOR>
struct Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits
```

Definition at line 56 of file VectorBase.h.

### 6.347.2 Member Typedef Documentation

#### 6.347.2.1 difference\_type

```
template<int RTYPE, bool na, typename VECTOR >
typedef R_xlen_t Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits::difference_type
```

Definition at line 60 of file VectorBase.h.

### 6.347.2.2 iterator\_category

```
template<int RTYPE, bool na, typename VECTOR >
typedef std::random_access_iterator_tag Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits::iterator_category
```

Definition at line 62 of file VectorBase.h.

### 6.347.2.3 pointer

```
template<int RTYPE, bool na, typename VECTOR >
typedef stored_type* Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits::pointer
```

Definition at line 59 of file VectorBase.h.

### 6.347.2.4 reference

```
template<int RTYPE, bool na, typename VECTOR >
typedef stored_type& Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits::reference
```

Definition at line 58 of file VectorBase.h.

### 6.347.2.5 value\_type

```
template<int RTYPE, bool na, typename VECTOR >
typedef stored_type Rcpp::VectorBase< RTYPE, na, VECTOR >::iter_traits::value_type
```

Definition at line 61 of file VectorBase.h.

The documentation for this struct was generated from the following file:

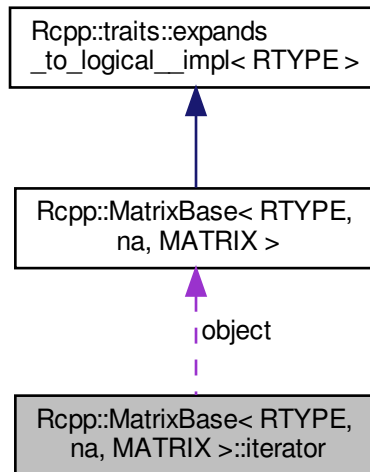
- [inst/include/Rcpp/vector/VectorBase.h](#)



## 6.348 Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator Class Reference

```
#include <MatrixBase.h>
```

Collaboration diagram for Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator:



### Public Types

- typedef [stored\\_type](#) [reference](#)
- typedef [stored\\_type](#) \* [pointer](#)
- typedef [R\\_xlen\\_t](#) [difference\\_type](#)
- typedef [stored\\_type](#) [value\\_type](#)
- typedef [std::random\\_access\\_iterator\\_tag](#) [iterator\\_category](#)

### Public Member Functions

- [iterator](#) (const [MatrixBase](#) &[object\\_](#), [R\\_xlen\\_t](#) [index\\_](#))
- [iterator](#) (const [iterator](#) &[other](#))
- [iterator](#) & [operator++](#) ()
- [iterator](#) [operator++](#) ([int](#))
- [iterator](#) & [operator--](#) ()
- [iterator](#) [operator--](#) ([int](#))
- [iterator](#) [operator+](#) ([difference\\_type](#) [n](#)) const
- [iterator](#) [operator-](#) ([difference\\_type](#) [n](#)) const
- [iterator](#) & [operator+=](#) ([difference\\_type](#) [n](#))
- [iterator](#) & [operator-=](#) ([difference\\_type](#) [n](#))

- [reference operator\\*](#) ()
- [pointer operator->](#) ()
- `bool operator== (const iterator &y) const`
- `bool operator!= (const iterator &y) const`
- `bool operator< (const iterator &other) const`
- `bool operator> (const iterator &other) const`
- `bool operator<= (const iterator &other) const`
- `bool operator>= (const iterator &other) const`
- `difference\_type operator- (const iterator &other) const`

## Private Member Functions

- `void update\_index (int index_)`
- `R_xlen_t index () const`

## Private Attributes

- `const MatrixBase & object`
- `int i`
- `int j`
- `int nr`
- `int nc`

### 6.348.1 Detailed Description

```
template<int RTYPE, bool na, typename MATRIX>
class Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator
```

Definition at line 83 of file MatrixBase.h.

### 6.348.2 Member Typedef Documentation

#### 6.348.2.1 `difference_type`

```
template<int RTYPE, bool na, typename MATRIX >
typedef R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::difference_type
```

Definition at line 87 of file MatrixBase.h.

### 6.348.2.2 iterator\_category

```
template<int RTYPE, bool na, typename MATRIX >  
typedef std::random_access_iterator_tag Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::iterator_category
```

Definition at line 89 of file MatrixBase.h.

### 6.348.2.3 pointer

```
template<int RTYPE, bool na, typename MATRIX >  
typedef stored_type* Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::pointer
```

Definition at line 86 of file MatrixBase.h.

### 6.348.2.4 reference

```
template<int RTYPE, bool na, typename MATRIX >  
typedef stored_type Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::reference
```

Definition at line 85 of file MatrixBase.h.

### 6.348.2.5 value\_type

```
template<int RTYPE, bool na, typename MATRIX >  
typedef stored_type Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::value_type
```

Definition at line 88 of file MatrixBase.h.

## 6.348.3 Constructor & Destructor Documentation

### 6.348.3.1 iterator() [1/2]

```
template<int RTYPE, bool na, typename MATRIX >
Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::iterator (
    const MatrixBase & object_,
    R_xlen_t index_ ) [inline]
```

Definition at line 91 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update\_index().

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-().

Here is the call graph for this function:



### 6.348.3.2 iterator() [2/2]

```
template<int RTYPE, bool na, typename MATRIX >
Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::iterator (
    const iterator & other ) [inline]
```

Definition at line 97 of file MatrixBase.h.

## 6.348.4 Member Function Documentation

### 6.348.4.1 index()

```
template<int RTYPE, bool na, typename MATRIX >
R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index ( ) const [inline], [private]
```

Definition at line 185 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator<(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator<=(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator>(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator>=().

### 6.348.4.2 operator"!=(())

```
template<int RTYPE, bool na, typename MATRIX >
bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator!= (
    const iterator & y ) const [inline]
```

Definition at line 154 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j.

### 6.348.4.3 operator\*()()

```
template<int RTYPE, bool na, typename MATRIX >
reference Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator* ( ) [inline]
```

Definition at line 144 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::object.

### 6.348.4.4 operator+()

```
template<int RTYPE, bool na, typename MATRIX >
iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+ (
    difference_type n ) const [inline]
```

Definition at line 128 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::iterator(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr.

Here is the call graph for this function:



**6.348.4.5 operator++()** [1/2]

```
template<int RTYPE, bool na, typename MATRIX >
iterator& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator++ ( ) [inline]
```

Definition at line 100 of file MatrixBase.h.

References [Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i](#), [Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j](#), and [Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr](#).

**6.348.4.6 operator++()** [2/2]

```
template<int RTYPE, bool na, typename MATRIX >
iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator++ (
    int ) [inline]
```

Definition at line 108 of file MatrixBase.h.

**6.348.4.7 operator+=()**

```
template<int RTYPE, bool na, typename MATRIX >
iterator& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+= (
    difference_type n ) [inline]
```

Definition at line 135 of file MatrixBase.h.

References [Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i](#), [Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j](#), [Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr](#), and [Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update\\_index\(\)](#).

Here is the call graph for this function:



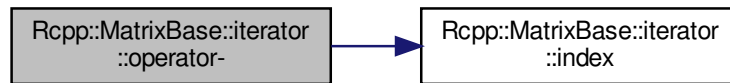
**6.348.4.8 operator-() [1/2]**

```
template<int RTYPE, bool na, typename MATRIX >
difference_type Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator- (
    const iterator & other ) const [inline]
```

Definition at line 170 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:

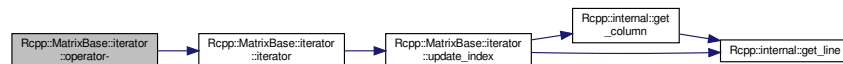
**6.348.4.9 operator-() [2/2]**

```
template<int RTYPE, bool na, typename MATRIX >
iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator- (
    difference_type n ) const [inline]
```

Definition at line 131 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::iterator(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr.

Here is the call graph for this function:



**6.348.4.10 operator--()** [1/2]

```
template<int RTYPE, bool na, typename MATRIX >
iterator& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-- ( ) [inline]
```

Definition at line 114 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >←  
::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr.

**6.348.4.11 operator--()** [2/2]

```
template<int RTYPE, bool na, typename MATRIX >
iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-- (
    int ) [inline]
```

Definition at line 122 of file MatrixBase.h.

**6.348.4.12 operator-=()**

```
template<int RTYPE, bool na, typename MATRIX >
iterator& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-= (
    difference_type n ) [inline]
```

Definition at line 139 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >←  
::iterator::j, Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr, and Rcpp::MatrixBase< RTYPE, na, MATRIX >←  
::iterator::update\_index().

Here is the call graph for this function:





#### 6.348.4.13 operator->()

```
template<int RTYPE, bool na, typename MATRIX >
pointer Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-> ( ) [inline]
```

Definition at line 147 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >←  
::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::object.

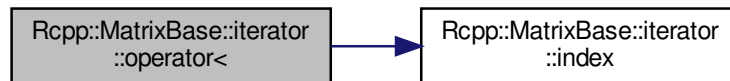
#### 6.348.4.14 operator<()

```
template<int RTYPE, bool na, typename MATRIX >
bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator< (
    const iterator & other ) const [inline]
```

Definition at line 157 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:



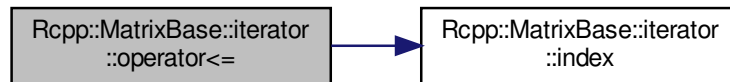
#### 6.348.4.15 operator<=()

```
template<int RTYPE, bool na, typename MATRIX >
bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator<= (
    const iterator & other ) const [inline]
```

Definition at line 163 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:



#### 6.348.4.16 operator==( )

```

template<int RTYPE, bool na, typename MATRIX >
bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator==(
    const iterator & y ) const [inline]
  
```

Definition at line 151 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, and Rcpp::MatrixBase< RTYPE, na, MATRIX >↔::iterator::j.

#### 6.348.4.17 operator>( )

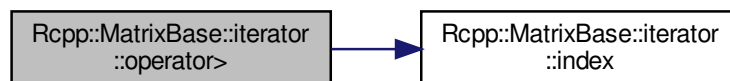
```

template<int RTYPE, bool na, typename MATRIX >
bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator> (
    const iterator & other ) const [inline]
  
```

Definition at line 160 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:



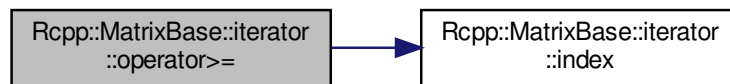
**6.348.4.18 operator>=()**

```
template<int RTYPE, bool na, typename MATRIX >
bool Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator>= (
    const iterator & other ) const [inline]
```

Definition at line 166 of file MatrixBase.h.

References Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index().

Here is the call graph for this function:

**6.348.4.19 update\_index()**

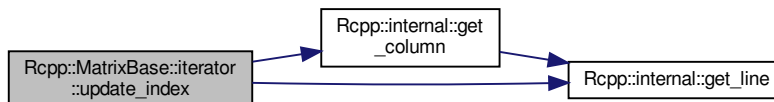
```
template<int RTYPE, bool na, typename MATRIX >
void Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update_index (
    int index_ ) [inline], [private]
```

Definition at line 180 of file MatrixBase.h.

References Rcpp::internal::get\_column(), Rcpp::internal::get\_line(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i, Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j, and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::iterator(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-().

Here is the call graph for this function:



## 6.348.5 Member Data Documentation

### 6.348.5.1 i

```
template<int RTYPE, bool na, typename MATRIX >
int Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::i [private]
```

Definition at line 177 of file MatrixBase.h.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator!==((), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator\*((), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+((), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator++(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+=((), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator--(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-==((), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator->(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator==((), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update\_index().

### 6.348.5.2 j

```
template<int RTYPE, bool na, typename MATRIX >
int Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::j [private]
```

Definition at line 177 of file MatrixBase.h.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator!==((), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator\*((), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+((), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator++(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+=((), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator--(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-==((), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator->(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator==((), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update\_index().

### 6.348.5.3 nc

```
template<int RTYPE, bool na, typename MATRIX >
int Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nc [private]
```

Definition at line 178 of file MatrixBase.h.

## 6.348.5.4 nr

```
template<int RTYPE, bool na, typename MATRIX >
int Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::nr [private]
```

Definition at line 178 of file MatrixBase.h.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::index(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator++(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator+=(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator--(), Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator-=(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::update\_index().

## 6.348.5.5 object

```
template<int RTYPE, bool na, typename MATRIX >
const MatrixBase& Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::object [private]
```

Definition at line 176 of file MatrixBase.h.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator\*(), and Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator::operator->().

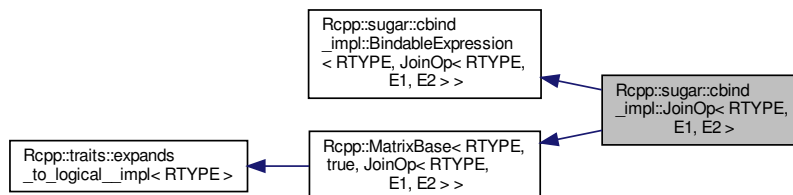
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixBase.h

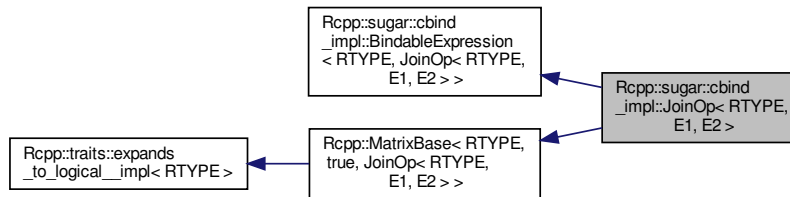
## 6.349 Rcpp::sugar::cbind\_impl::JoinOp&lt; RTYPE, E1, E2 &gt; Class Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >:



Collaboration diagram for `Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >`:



## Public Types

- typedef `cbind_storage_type< RTYPE >::type stored_type`

## Public Member Functions

- `JoinOp` (const `BindableExpression< RTYPE, E1 >` &e1\_, const `BindableExpression< RTYPE, E2 >` &e2\_)
- `R_xlen_t size` () const
- `R_xlen_t nrow` () const
- `R_xlen_t ncol` () const
- `stored_type operator[]` (R\_xlen\_t i) const
- `stored_type operator()` (R\_xlen\_t i, R\_xlen\_t j) const

## Private Attributes

- const E1 & `e1`
- const E2 & `e2`

## Additional Inherited Members

### 6.349.1 Detailed Description

```

template<int RTYPE, typename E1, typename E2>
class Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >

```

Definition at line 166 of file `cbind.h`.

### 6.349.2 Member Typedef Documentation

### 6.349.2.1 stored\_type

```
template<int RTYPE, typename E1 , typename E2 >
typedef cbind_storage_type<RTYPE>::type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::stored_type
```

Definition at line 170 of file cbind.h.

## 6.349.3 Constructor & Destructor Documentation

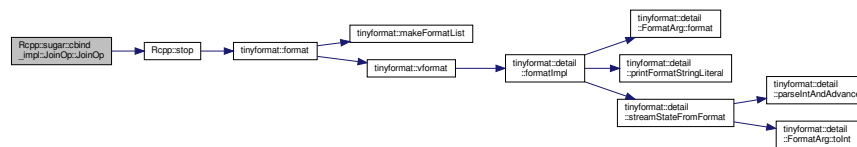
### 6.349.3.1 JoinOp()

```
template<int RTYPE, typename E1 , typename E2 >
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::JoinOp (
    const BindableExpression< RTYPE, E1 > & e1_,
    const BindableExpression< RTYPE, E2 > & e2_ ) [inline]
```

Definition at line 177 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e1, Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e2, and Rcpp::stop().

Here is the call graph for this function:



## 6.349.4 Member Function Documentation

### 6.349.4.1 ncol()

```
template<int RTYPE, typename E1 , typename E2 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::ncol ( ) const [inline]
```

Definition at line 195 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e1, and Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e2.

### 6.349.4.2 nrow()

```
template<int RTYPE, typename E1 , typename E2 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::nrow ( ) const [inline]
```

Definition at line 193 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e1.

Referenced by Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::operator(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::operator(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::operator(), and Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::operator().

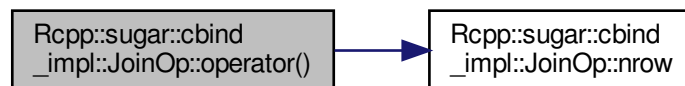
### 6.349.4.3 operator()()

```
template<int RTYPE, typename E1 , typename E2 >
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::operator() (
    R_xlen_t i,
    R_xlen_t j ) const [inline]
```

Definition at line 201 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::nrow().

Here is the call graph for this function:



### 6.349.4.4 operator[]()

```
template<int RTYPE, typename E1 , typename E2 >
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 197 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e1, and Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e2.



### 6.349.4.5 size()

```
template<int RTYPE, typename E1 , typename E2 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::size ( ) const [inline]
```

Definition at line 191 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e1, and Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e2.

## 6.349.5 Member Data Documentation

### 6.349.5.1 e1

```
template<int RTYPE, typename E1 , typename E2 >
const E1& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1 [private]
```

Definition at line 173 of file cbind.h.

Referenced by Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::JoinOp(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::ncol(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::ncol(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::nrow(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::nrow(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::operator[](), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::operator[](), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type > >, E2 >::operator[](), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type > >, ScalarBindable< typename scalar< RTYPE >::type > >::operator[](), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::size(), and Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::size().

### 6.349.5.2 e2

```
template<int RTYPE, typename E1 , typename E2 >
const E2& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2 [private]
```

Definition at line 174 of file cbind.h.

Referenced by Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::JoinOp(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::ncol(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type > >, E2 >::ncol(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type > >, ScalarBindable< typename scalar< RTYPE >::type > >::ncol(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type > >, E2 >::nrow(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type > >, ScalarBindable< typename scalar< RTYPE >::type > >::nrow(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::operator[](), Rcpp::sugar::cbind\_impl::JoinOp<

RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::operator[ ](), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::operator[ ](), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::operator[ ](), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::size(), Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::size(), and Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::size().

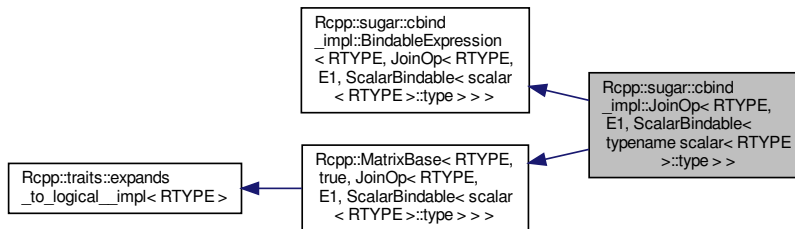
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

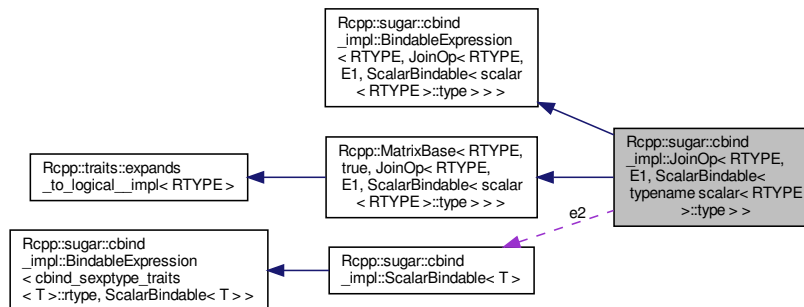
### 6.350 Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > > Class Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >:



Collaboration diagram for Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >:



## Public Types

- typedef [cbind\\_storage\\_type](#)< RTYPE >::type [stored\\_type](#)
- typedef [ScalarBindable](#)< typename [scalar](#)< RTYPE >::type > [E2](#)

## Public Member Functions

- [JoinOp](#) (const [BindableExpression](#)< RTYPE, E1 > &e1\_, const [BindableExpression](#)< RTYPE, E2 > &e2\_)
- [R\\_xlen\\_t size](#) () const
- [R\\_xlen\\_t nrow](#) () const
- [R\\_xlen\\_t ncol](#) () const
- [stored\\_type operator\[\]](#) (R\_xlen\_t i) const
- [stored\\_type operator\(\)](#) (R\_xlen\_t i, R\_xlen\_t j) const

## Private Attributes

- const E1 & [e1](#)
- const [E2](#) & [e2](#)

## Additional Inherited Members

### 6.350.1 Detailed Description

```
template<int RTYPE, typename E1>  
class Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >
```

Definition at line 209 of file cbind.h.

### 6.350.2 Member Typedef Documentation

#### 6.350.2.1 E2

```
template<int RTYPE, typename E1 >  
typedef ScalarBindable<typename scalar<RTYPE>::type> Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1,  
ScalarBindable< typename scalar< RTYPE >::type > >::E2
```

Definition at line 218 of file cbind.h.

### 6.350.2.2 stored\_type

```
template<int RTYPE, typename E1 >
typedef cbind_storage_type<RTYPE>::type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable<
typename scalar< RTYPE >::type > >::stored_type
```

Definition at line 217 of file cbind.h.

## 6.350.3 Constructor & Destructor Documentation

### 6.350.3.1 JoinOp()

```
template<int RTYPE, typename E1 >
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >↔
::JoinOp (
    const BindableExpression< RTYPE, E1 > & e1_,
    const BindableExpression< RTYPE, E2 > & e2_ ) [inline]
```

Definition at line 225 of file cbind.h.

## 6.350.4 Member Function Documentation

### 6.350.4.1 ncol()

```
template<int RTYPE, typename E1 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >↔
::type > >::ncol ( ) const [inline]
```

Definition at line 234 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e1.

### 6.350.4.2 nrow()

```
template<int RTYPE, typename E1 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >↔
::type > >::nrow ( ) const [inline]
```

Definition at line 232 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e1.

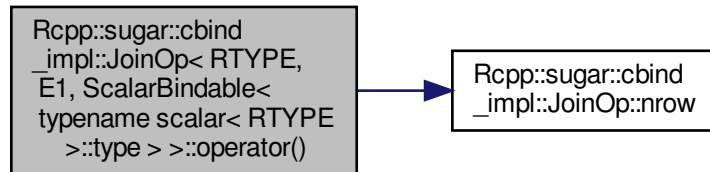
### 6.350.4.3 operator>()

```
template<int RTYPE, typename E1 >  
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE  
>::type > >::operator() (  
    R_xlen_t i,  
    R_xlen_t j ) const [inline]
```

Definition at line 240 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::nrow().

Here is the call graph for this function:



### 6.350.4.4 operator[]()

```
template<int RTYPE, typename E1 >  
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE  
>::type > >::operator[] (  
    R_xlen_t i ) const [inline]
```

Definition at line 236 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e1, and Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e2.

### 6.350.4.5 size()

```
template<int RTYPE, typename E1 >  
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >↔  
::type > >::size ( ) const [inline]
```

Definition at line 230 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e1.

## 6.350.5 Member Data Documentation

### 6.350.5.1 e1

```
template<int RTYPE, typename E1 >
const E1& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::e1 [private]
```

Definition at line 221 of file cbind.h.

### 6.350.5.2 e2

```
template<int RTYPE, typename E1 >
const E2& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >::e2 [private]
```

Definition at line 222 of file cbind.h.

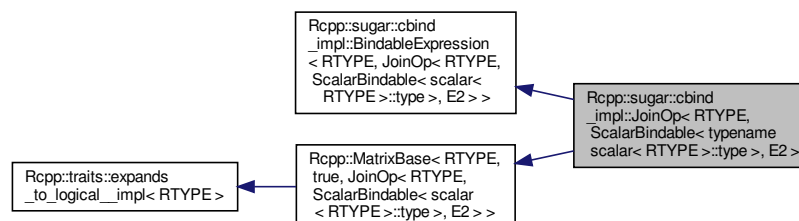
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

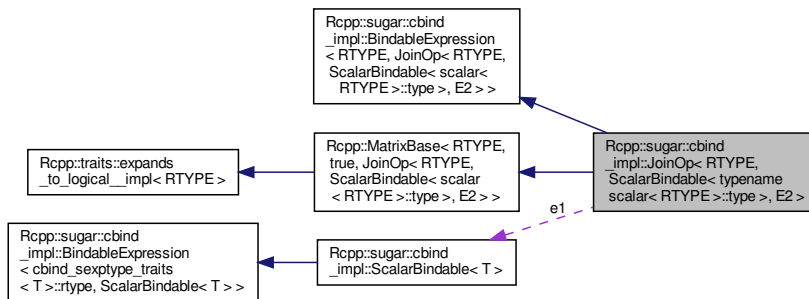
## 6.351 Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 > Class Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >:



Collaboration diagram for Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >:



## Public Types

- typedef [cbind\\_storage\\_type](#)< RTYPE >::type [stored\\_type](#)
- typedef [ScalarBindable](#)< typename [scalar](#)< RTYPE >::type > [E1](#)

## Public Member Functions

- [JoinOp](#) (const [BindableExpression](#)< RTYPE, [E1](#) > &e1\_, const [BindableExpression](#)< RTYPE, [E2](#) > &e2\_)
- [R\\_xlen\\_t size](#) () const
- [R\\_xlen\\_t nrow](#) () const
- [R\\_xlen\\_t ncol](#) () const
- [stored\\_type operator\[\]](#) ([R\\_xlen\\_t](#) i) const
- [stored\\_type operator\(\)](#) ([R\\_xlen\\_t](#) i, [R\\_xlen\\_t](#) j) const

## Private Attributes

- const [E1](#) & [e1](#)
- const [E2](#) & [e2](#)

## Additional Inherited Members

### 6.351.1 Detailed Description

```
template<int RTYPE, typename E2>
class Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >
```

Definition at line 248 of file cbind.h.

## 6.351.2 Member Typedef Documentation

### 6.351.2.1 E1

```
template<int RTYPE, typename E2 >
typedef ScalarBindable<typename scalar<RTYPE>::type> Rcpp::sugar::cbind_impl::JoinOp< RTYPE,
ScalarBindable< typename scalar< RTYPE >::type >, E2 >::E1
```

Definition at line 257 of file cbind.h.

### 6.351.2.2 stored\_type

```
template<int RTYPE, typename E2 >
typedef cbind_storage_type<RTYPE>::type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable<
typename scalar< RTYPE >::type >, E2 >::stored_type
```

Definition at line 256 of file cbind.h.

## 6.351.3 Constructor & Destructor Documentation

### 6.351.3.1 JoinOp()

```
template<int RTYPE, typename E2 >
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >↔
::JoinOp (
    const BindableExpression< RTYPE, E1 > & e1_,
    const BindableExpression< RTYPE, E2 > & e2_ ) [inline]
```

Definition at line 264 of file cbind.h.

## 6.351.4 Member Function Documentation



#### 6.351.4.1 ncol()

```
template<int RTYPE, typename E2 >  
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type  
>, E2 >::ncol ( ) const [inline]
```

Definition at line 273 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e2.

#### 6.351.4.2 nrow()

```
template<int RTYPE, typename E2 >  
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type  
>, E2 >::nrow ( ) const [inline]
```

Definition at line 271 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e2.

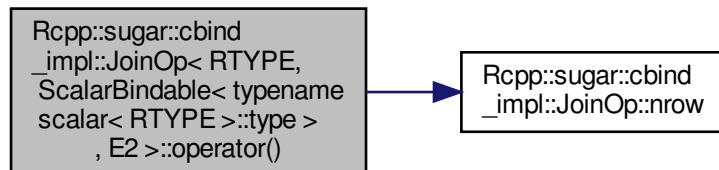
#### 6.351.4.3 operator>()()

```
template<int RTYPE, typename E2 >  
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >←  
::type >, E2 >::operator() (   
    R_xlen_t i,  
    R_xlen_t j ) const [inline]
```

Definition at line 279 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::nrow().

Here is the call graph for this function:



#### 6.351.4.4 operator[]()

```
template<int RTYPE, typename E2 >
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 275 of file cbind.h.

References `Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e1`, and `Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2`.

#### 6.351.4.5 size()

```
template<int RTYPE, typename E2 >
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::size ( ) const [inline]
```

Definition at line 269 of file cbind.h.

References `Rcpp::sugar::cbind_impl::JoinOp< RTYPE, E1, E2 >::e2`.

### 6.351.5 Member Data Documentation

#### 6.351.5.1 e1

```
template<int RTYPE, typename E2 >
const E1& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::e1 [private]
```

Definition at line 260 of file cbind.h.

#### 6.351.5.2 e2

```
template<int RTYPE, typename E2 >
const E2& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >::e2 [private]
```

Definition at line 261 of file cbind.h.

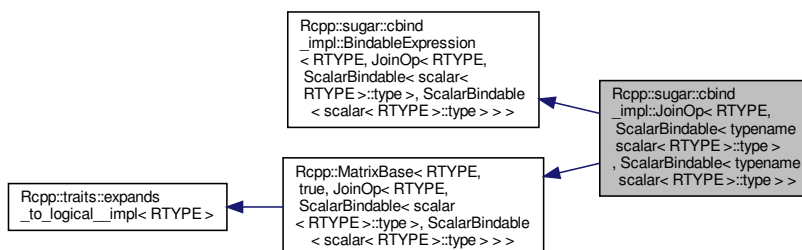
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/cbind.h`

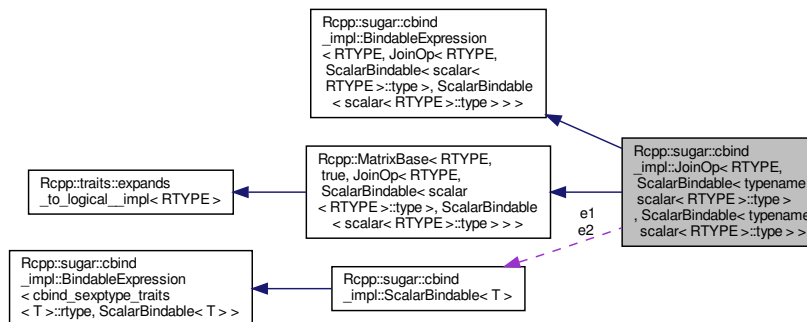
## 6.352 Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > > Class Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >:



Collaboration diagram for Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >:



### Public Types

- typedef [cbind\\_storage\\_type< RTYPE >::type](#) [stored\\_type](#)
- typedef [ScalarBindable< typename scalar< RTYPE >::type > E1](#)
- typedef [ScalarBindable< typename scalar< RTYPE >::type > E2](#)

## Public Member Functions

- [JoinOp](#) (const [BindableExpression](#)< RTYPE, [E1](#) > &e1\_, const [BindableExpression](#)< RTYPE, [E2](#) > &e2\_)
- [R\\_xlen\\_t size](#) () const
- [R\\_xlen\\_t nrow](#) () const
- [R\\_xlen\\_t ncol](#) () const
- [stored\\_type operator\[\]](#) (R\_xlen\_t i) const
- [stored\\_type operator\(\)](#) (R\_xlen\_t i, R\_xlen\_t j) const

## Private Attributes

- const [E1](#) & [e1](#)
- const [E2](#) & [e2](#)

## Additional Inherited Members

### 6.352.1 Detailed Description

```
template<int RTYPE>
class Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename
scalar< RTYPE >::type > >
```

Definition at line 287 of file cbind.h.

### 6.352.2 Member Typedef Documentation

#### 6.352.2.1 E1

```
template<int RTYPE>
typedef ScalarBindable<typename scalar<RTYPE>::type> Rcpp::sugar::cbind\_impl::JoinOp< RTYPE,
ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >↔
::type > >::E1
```

Definition at line 299 of file cbind.h.

#### 6.352.2.2 E2

```
template<int RTYPE>
typedef ScalarBindable<typename scalar<RTYPE>::type> Rcpp::sugar::cbind\_impl::JoinOp< RTYPE,
ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >↔
::type > >::E2
```

Definition at line 300 of file cbind.h.

### 6.352.2.3 stored\_type

```
template<int RTYPE>
typedef cbind_storage_type<RTYPE>::type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable<
typename scalar< RTYPE >::type >, ScalarBindable< typename scalar< RTYPE >::type > >::stored_type
```

Definition at line 298 of file cbind.h.

## 6.352.3 Constructor & Destructor Documentation

### 6.352.3.1 JoinOp()

```
template<int RTYPE>
Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable<
typename scalar< RTYPE >::type > >::JoinOp (
    const BindableExpression< RTYPE, E1 > & e1_,
    const BindableExpression< RTYPE, E2 > & e2_ ) [inline]
```

Definition at line 307 of file cbind.h.

## 6.352.4 Member Function Documentation

### 6.352.4.1 ncol()

```
template<int RTYPE>
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, ScalarBindable< typename scalar< RTYPE >::type > >::ncol ( ) const [inline]
```

Definition at line 316 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e2.

### 6.352.4.2 nrow()

```
template<int RTYPE>
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, ScalarBindable< typename scalar< RTYPE >::type > >::nrow ( ) const [inline]
```

Definition at line 314 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e2.

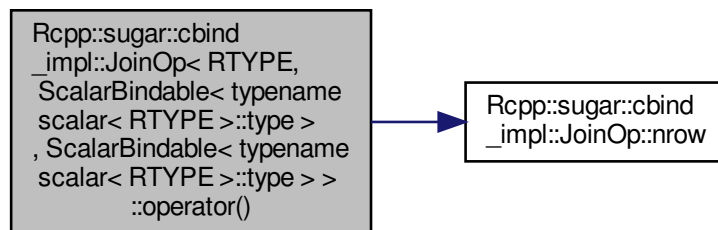
### 6.352.4.3 operator()

```
template<int RTYPE>
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >↔
::type >, ScalarBindable< typename scalar< RTYPE >::type > >::operator() (
    R_xlen_t i,
    R_xlen_t j ) const [inline]
```

Definition at line 322 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::nrow().

Here is the call graph for this function:



### 6.352.4.4 operator[]()

```
template<int RTYPE>
stored_type Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >↔
::type >, ScalarBindable< typename scalar< RTYPE >::type > >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 318 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e1, and Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e2.

### 6.352.4.5 size()

```
template<int RTYPE>
R_xlen_t Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, ScalarBindable< typename scalar< RTYPE >::type > >::size ( ) const [inline]
```

Definition at line 312 of file cbind.h.

References Rcpp::sugar::cbind\_impl::JoinOp< RTYPE, E1, E2 >::e2.

## 6.352.5 Member Data Documentation

### 6.352.5.1 e1

```
template<int RTYPE>
const E1& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, ScalarBindable< typename scalar< RTYPE >::type > >::e1 [private]
```

Definition at line 303 of file cbind.h.

### 6.352.5.2 e2

```
template<int RTYPE>
const E2& Rcpp::sugar::cbind_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type
>, ScalarBindable< typename scalar< RTYPE >::type > >::e2 [private]
```

Definition at line 304 of file cbind.h.

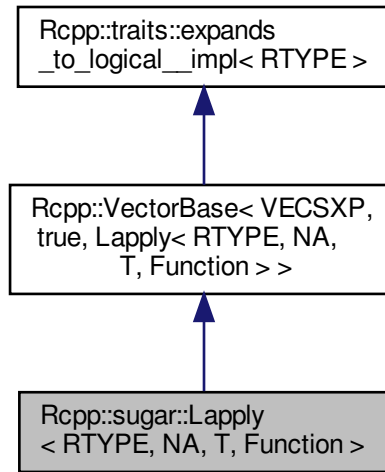
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[cbind.h](#)

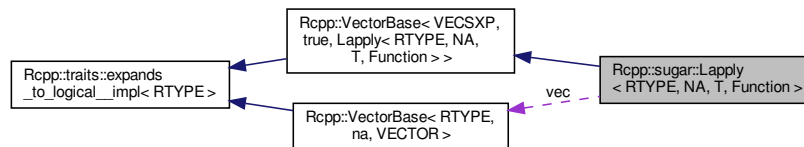
## 6.353 Rcpp::sugar::Lapply< RTYPE, NA, T, Function > Class Template Reference

```
#include <lapply.h>
```

Inheritance diagram for `Rcpp::sugar::Lapply< RTYPE, NA, T, Function >`:



Collaboration diagram for `Rcpp::sugar::Lapply< RTYPE, NA, T, Function >`:



## Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC`
- typedef `::Rcpp::traits::result_of< Function >::type` `result_type`

## Public Member Functions

- `Lapply` (const `VEC` &vec\_, `Function` fun\_)
- `SEXP operator[]` (`R_xlen_t` i) const
- `R_xlen_t size` () const



## Private Attributes

- const [VEC](#) & [vec](#)
- [Function](#) [fun](#)

### 6.353.1 Detailed Description

```
template<int RTYPE, bool NA, typename T, typename Function>
class Rcpp::sugar::Lapply< RTYPE, NA, T, Function >
```

Definition at line 29 of file `lapply.h`.

### 6.353.2 Member Typedef Documentation

#### 6.353.2.1 result\_type

```
template<int RTYPE, bool NA, typename T , typename Function >
typedef ::Rcpp::traits::result_of<Function>::type Rcpp::sugar::Lapply< RTYPE, NA, T, Function
>::result_type
```

Definition at line 36 of file `lapply.h`.

#### 6.353.2.2 VEC

```
template<int RTYPE, bool NA, typename T , typename Function >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::VEC
```

Definition at line 35 of file `lapply.h`.

### 6.353.3 Constructor & Destructor Documentation

#### 6.353.3.1 Lapply()

```
template<int RTYPE, bool NA, typename T , typename Function >
Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::Lapply (
    const VEC & vec\_,
    Function fun\_ ) [inline]
```

Definition at line 38 of file `lapply.h`.

## 6.353.4 Member Function Documentation

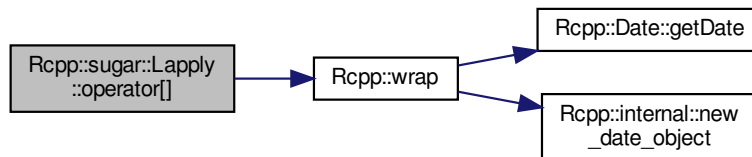
### 6.353.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T , typename Function >
SEXP Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 41 of file lapply.h.

References `Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::fun`, `Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::vec`, and `Rcpp::wrap()`.

Here is the call graph for this function:



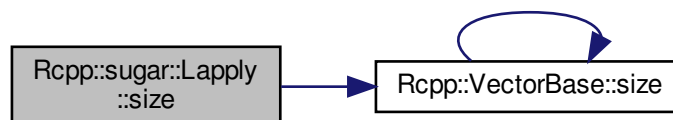
### 6.353.4.2 size()

```
template<int RTYPE, bool NA, typename T , typename Function >
R_xlen_t Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::size ( ) const [inline]
```

Definition at line 44 of file lapply.h.

References `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`, and `Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::vec`.

Here is the call graph for this function:



## 6.353.5 Member Data Documentation

### 6.353.5.1 fun

```
template<int RTYPE, bool NA, typename T , typename Function >
Function Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::fun [private]
```

Definition at line 48 of file lapply.h.

Referenced by Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::operator[]().

### 6.353.5.2 vec

```
template<int RTYPE, bool NA, typename T , typename Function >
const VEC& Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::vec [private]
```

Definition at line 47 of file lapply.h.

Referenced by Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::operator[](), and Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::size().

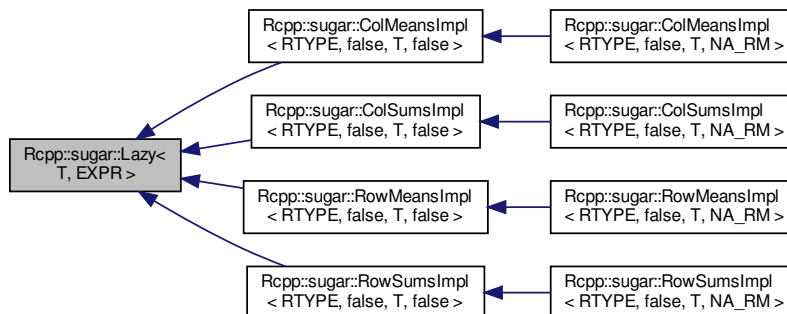
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/lapply.h](#)

## 6.354 Rcpp::sugar::Lazy< T, EXPR > Class Template Reference

```
#include <Lazy.h>
```

Inheritance diagram for Rcpp::sugar::Lazy< T, EXPR >:



## Public Member Functions

- [operator T \(\) const](#)

### 6.354.1 Detailed Description

```
template<typename T, typename Expr>
class Rcpp::sugar::Lazy< T, Expr >
```

Definition at line 29 of file Lazy.h.

### 6.354.2 Member Function Documentation

#### 6.354.2.1 operator T()

```
template<typename T , typename Expr >
Rcpp::sugar::Lazy< T, Expr >::operator T ( ) const [inline]
```

Definition at line 31 of file Lazy.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/Lazy.h](#)

## 6.355 Rcpp::internal::LazyVector< VECTOR > Class Template Reference

```
#include <LazyVector.h>
```

### Public Types

- typedef VECTOR::r\_type [r\\_type](#)
- typedef [Rcpp::traits::storage\\_type](#)< r\_type::value >::type [stored\\_type](#)

### Public Member Functions

- [LazyVector](#) (const VECTOR &vec\_)
- [stored\\_type operator\[\]](#) (R\_xlen\_t i) const

## Private Attributes

- const VECTOR & [vec](#)
- [R\\_xlen\\_t](#) n
- [std::vector< stored\\_type >](#) data
- [std::vector< bool >](#) known

### 6.355.1 Detailed Description

```
template<typename VECTOR>
class Rcpp::internal::LazyVector< VECTOR >
```

Definition at line 29 of file LazyVector.h.

### 6.355.2 Member Typedef Documentation

#### 6.355.2.1 r\_type

```
template<typename VECTOR >
typedef VECTOR::r_type Rcpp::internal::LazyVector< VECTOR >::r_type
```

Definition at line 31 of file LazyVector.h.

#### 6.355.2.2 stored\_type

```
template<typename VECTOR >
typedef Rcpp::traits::storage_type< r_type::value >::type Rcpp::internal::LazyVector< VECTOR >↔
::stored_type
```

Definition at line 32 of file LazyVector.h.

### 6.355.3 Constructor & Destructor Documentation

### 6.355.3.1 LazyVector()

```
template<typename VECTOR >  
Rcpp::internal::LazyVector< VECTOR >::LazyVector (   
    const VECTOR & vec_ ) [inline]
```

Definition at line 34 of file LazyVector.h.

## 6.355.4 Member Function Documentation

### 6.355.4.1 operator[]()

```
template<typename VECTOR >  
stored_type Rcpp::internal::LazyVector< VECTOR >::operator[] (   
    R_xlen_t i ) const [inline]
```

Definition at line 36 of file LazyVector.h.

References `Rcpp::internal::LazyVector< VECTOR >::data`, `Rcpp::internal::LazyVector< VECTOR >::known`, and `Rcpp::internal::LazyVector< VECTOR >::vec`.

## 6.355.5 Member Data Documentation

### 6.355.5.1 data

```
template<typename VECTOR >  
std::vector<stored_type> Rcpp::internal::LazyVector< VECTOR >::data [mutable], [private]
```

Definition at line 50 of file LazyVector.h.

Referenced by `Rcpp::internal::LazyVector< VECTOR >::operator[]()`.

### 6.355.5.2 known

```
template<typename VECTOR >  
std::vector<bool> Rcpp::internal::LazyVector< VECTOR >::known [mutable], [private]
```

Definition at line 51 of file LazyVector.h.

Referenced by `Rcpp::internal::LazyVector< VECTOR >::operator[]()`.

## 6.355.5.3 n

```
template<typename VECTOR >
R_xlen_t Rcpp::internal::LazyVector< VECTOR >::n [private]
```

Definition at line 49 of file LazyVector.h.

## 6.355.5.4 vec

```
template<typename VECTOR >
const VECTOR& Rcpp::internal::LazyVector< VECTOR >::vec [private]
```

Definition at line 48 of file LazyVector.h.

Referenced by Rcpp::internal::LazyVector< VECTOR >::operator[](), and Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::operator[]().

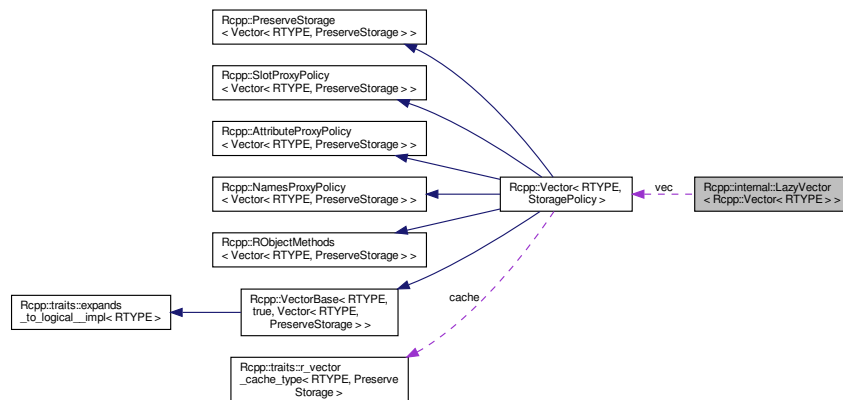
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/LazyVector.h

## 6.356 Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > > Class Template Reference

```
#include <LazyVector.h>
```

Collaboration diagram for Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >:



## Public Types

- typedef [Rcpp::Vector](#)< RTYPE > [VECTOR](#)
- typedef [VECTOR::Proxy](#) [Proxy](#)

## Public Member Functions

- [LazyVector](#) (const [VECTOR](#) &vec\_)
- [Proxy operator\[\]](#) (R\_xlen\_t i) const

## Private Attributes

- const [VECTOR](#) & [vec](#)

### 6.356.1 Detailed Description

```
template<int RTYPE>
class Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >
```

Definition at line 55 of file LazyVector.h.

### 6.356.2 Member Typedef Documentation

#### 6.356.2.1 Proxy

```
template<int RTYPE>
typedef VECTOR::Proxy Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::Proxy
```

Definition at line 58 of file LazyVector.h.

#### 6.356.2.2 VECTOR

```
template<int RTYPE>
typedef Rcpp::Vector<RTYPE> Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::VECTOR
```

Definition at line 57 of file LazyVector.h.



## 6.356.3 Constructor & Destructor Documentation

### 6.356.3.1 LazyVector()

```
template<int RTYPE>
Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::LazyVector (
    const VECTOR & vec_ ) [inline]
```

Definition at line 60 of file LazyVector.h.

## 6.356.4 Member Function Documentation

### 6.356.4.1 operator[]()

```
template<int RTYPE>
Proxy Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 61 of file LazyVector.h.

References Rcpp::internal::LazyVector< VECTOR >::vec.

## 6.356.5 Member Data Documentation

### 6.356.5.1 vec

```
template<int RTYPE>
const VECTOR& Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >::vec [private]
```

Definition at line 64 of file LazyVector.h.

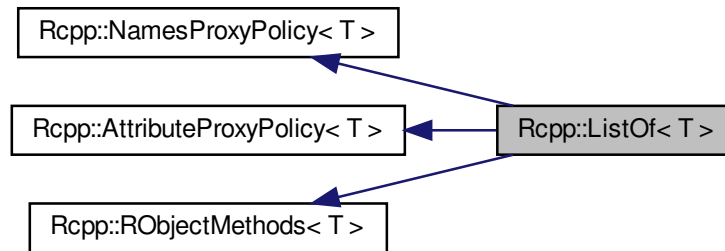
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/LazyVector.h

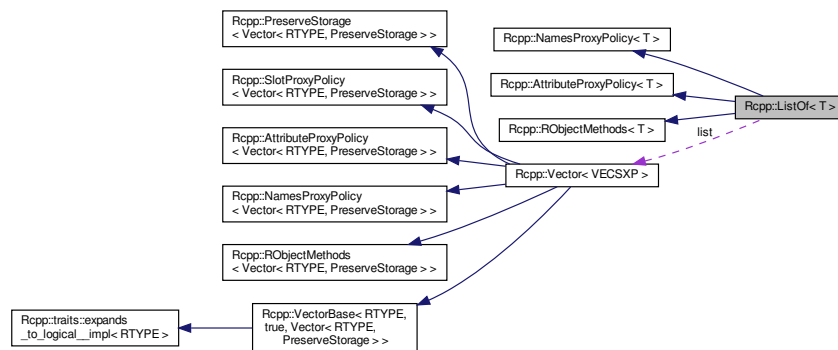
## 6.357 Rcpp::ListOf< T > Class Template Reference

```
#include <ListOf.h>
```

Inheritance diagram for Rcpp::ListOf< T >:



Collaboration diagram for Rcpp::ListOf< T >:



### Public Types

- typedef [traits::r\\_vector\\_iterator< VECSXP >::type](#) `iterator`
- typedef [traits::r\\_vector\\_const\\_iterator< VECSXP >::type](#) `const_iterator`

### Public Member Functions

- [ListOf\(\)](#)
- [ListOf\(SEXP data\\_\)](#)

- `template<typename U >`  
`ListOf (const U &data_)`
- `ListOf (const ListOf &other)`
- `ListOf & operator= (const ListOf &other)`
- `template<typename U >`  
`ListOf & operator= (const U &other)`
- `ChildVector< T > operator[] (R_xlen_t i)`
- `const ChildVector< T > operator[] (R_xlen_t i) const`
- `ChildVector< T > operator[] (const std::string &str)`
- `const ChildVector< T > operator[] (const std::string &str) const`
- `iterator begin ()`
- `iterator end ()`
- `const_iterator begin () const`
- `const_iterator end () const`
- `R_xlen_t size () const`
- `List get () const`
- `operator SEXP () const`
- `operator List () const`

## Private Attributes

- [List list](#)

### 6.357.1 Detailed Description

```
template<typename T>
class Rcpp::ListOf< T >
```

Definition at line 26 of file ListOf.h.

### 6.357.2 Member Typedef Documentation

#### 6.357.2.1 const\_iterator

```
template<typename T >
typedef traits::r_vector_const_iterator<VECSXP>::type Rcpp::ListOf< T >::const_iterator
```

Definition at line 34 of file ListOf.h.

### 6.357.2.2 iterator

```
template<typename T >
typedef traits::r_vector_iterator<VECSXP>::type Rcpp::ListOf< T >::iterator
```

Definition at line 33 of file ListOf.h.

## 6.357.3 Constructor & Destructor Documentation

### 6.357.3.1 ListOf() [1/4]

```
template<typename T >
Rcpp::ListOf< T >::ListOf ( ) [inline]
```

Definition at line 36 of file ListOf.h.

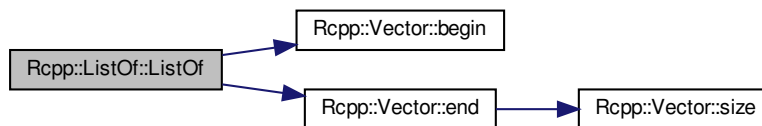
### 6.357.3.2 ListOf() [2/4]

```
template<typename T >
Rcpp::ListOf< T >::ListOf (
    SEXP data_ ) [inline]
```

Definition at line 38 of file ListOf.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, and `Rcpp::ListOf< T >::list`.

Here is the call graph for this function:



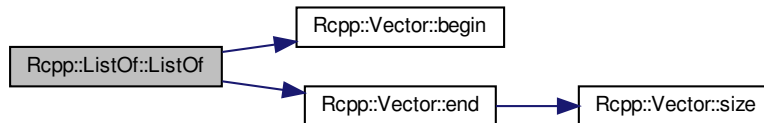
### 6.357.3.3 ListOf() [3/4]

```
template<typename T >
template<typename U >
Rcpp::ListOf< T >::ListOf (
    const U & data_ ) [inline]
```

Definition at line 43 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:



### 6.357.3.4 ListOf() [4/4]

```
template<typename T >
Rcpp::ListOf< T >::ListOf (
    const ListOf< T > & other ) [inline]
```

Definition at line 47 of file ListOf.h.

## 6.357.4 Member Function Documentation

### 6.357.4.1 begin() [1/2]

```
template<typename T >  
iterator Rcpp::ListOf< T >::begin ( ) [inline]
```

Definition at line 82 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:



### 6.357.4.2 begin() [2/2]

```
template<typename T >  
const_iterator Rcpp::ListOf< T >::begin ( ) const [inline]
```

Definition at line 90 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:



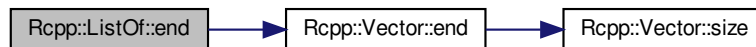
### 6.357.4.3 end() [1/2]

```
template<typename T >  
iterator Rcpp::ListOf< T >::end ( ) [inline]
```

Definition at line 86 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::end(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:



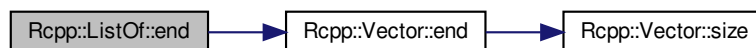
### 6.357.4.4 end() [2/2]

```
template<typename T >  
const_iterator Rcpp::ListOf< T >::end ( ) const [inline]
```

Definition at line 94 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::end(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:



### 6.357.4.5 get()

```
template<typename T >  
List Rcpp::ListOf< T >::get ( ) const [inline]
```

Definition at line 102 of file ListOf.h.

References Rcpp::ListOf< T >::list.

Referenced by Rcpp::lapply(), and Rcpp::sapply().

### 6.357.4.6 operator List()

```
template<typename T >
Rcpp::ListOf< T >::operator List ( ) const [inline]
```

Definition at line 108 of file ListOf.h.

References Rcpp::ListOf< T >::list.

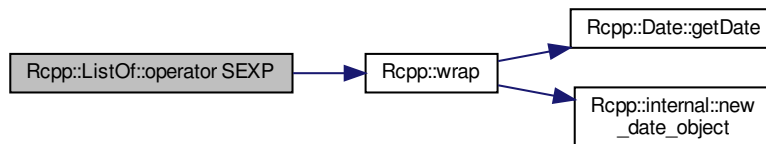
### 6.357.4.7 operator SEXP()

```
template<typename T >
Rcpp::ListOf< T >::operator SEXP ( ) const [inline]
```

Definition at line 107 of file ListOf.h.

References Rcpp::ListOf< T >::list, and Rcpp::wrap().

Here is the call graph for this function:



### 6.357.4.8 operator=() [1/2]

```
template<typename T >
ListOf& Rcpp::ListOf< T >::operator= (
    const ListOf< T > & other ) [inline]
```

Definition at line 49 of file ListOf.h.

References Rcpp::ListOf< T >::list.



## 6.357.4.9 operator=( ) [2/2]

```
template<typename T >
template<typename U >
ListOf& Rcpp::ListOf< T >::operator= (
    const U & other ) [inline]
```

Definition at line 57 of file ListOf.h.

References Rcpp::ListOf< T >::list.

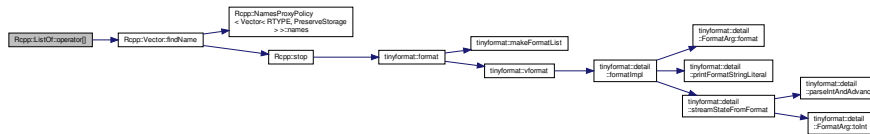
## 6.357.4.10 operator[]( ) [1/4]

```
template<typename T >
ChildVector<T> Rcpp::ListOf< T >::operator[] (
    const std::string & str ) [inline]
```

Definition at line 72 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::findName(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:



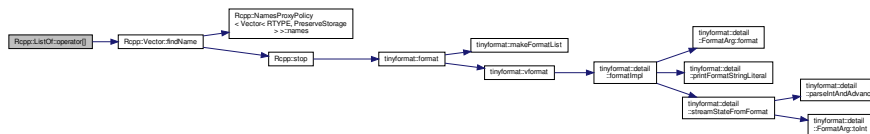
## 6.357.4.11 operator[]( ) [2/4]

```
template<typename T >
const ChildVector<T> Rcpp::ListOf< T >::operator[] (
    const std::string & str ) const [inline]
```

Definition at line 76 of file ListOf.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::findName(), and Rcpp::ListOf< T >::list.

Here is the call graph for this function:



**6.357.4.12 operator[]() [3/4]**

```
template<typename T >
ChildVector<T> Rcpp::ListOf< T >::operator[] (
    R_xlen_t i ) [inline]
```

Definition at line 64 of file ListOf.h.

References Rcpp::ListOf< T >::list.

**6.357.4.13 operator[]() [4/4]**

```
template<typename T >
const ChildVector<T> Rcpp::ListOf< T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 68 of file ListOf.h.

References Rcpp::ListOf< T >::list.

**6.357.4.14 size()**

```
template<typename T >
R_xlen_t Rcpp::ListOf< T >::size ( ) const [inline]
```

Definition at line 98 of file ListOf.h.

References Rcpp::ListOf< T >::list, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:

**6.357.5 Member Data Documentation**

## 6.357.5.1 list

```
template<typename T >
List Rcpp::ListOf< T >::list [private]
```

Definition at line 112 of file ListOf.h.

Referenced by Rcpp::ListOf< T >::begin(), Rcpp::ListOf< T >::end(), Rcpp::ListOf< T >::get(), Rcpp::ListOf< T >::ListOf(), Rcpp::ListOf< T >::operator List(), Rcpp::ListOf< T >::operator SEXP(), Rcpp::ListOf< T >::operator=(), Rcpp::ListOf< T >::operator[](), and Rcpp::ListOf< T >::size().

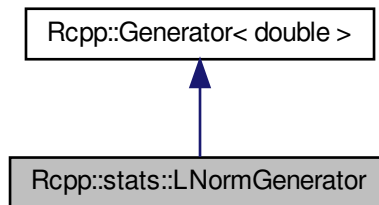
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/ListOf.h

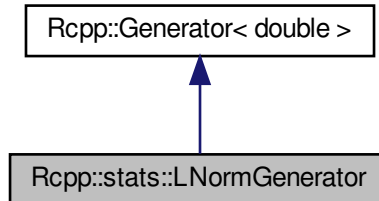
## 6.358 Rcpp::stats::LNormGenerator Class Reference

```
#include <rlnorm.h>
```

Inheritance diagram for Rcpp::stats::LNormGenerator:



Collaboration diagram for Rcpp::stats::LNormGenerator:



## Public Member Functions

- [LNormGenerator](#) (double meanlog\_=0.0, double sdlog\_=1.0)
- double [operator\(\)](#) () const

## Private Attributes

- double [meanlog](#)
- double [sdlog](#)

## Additional Inherited Members

### 6.358.1 Detailed Description

Definition at line 28 of file rlnorm.h.

### 6.358.2 Constructor & Destructor Documentation

#### 6.358.2.1 LNormGenerator()

```
Rcpp::stats::LNormGenerator::LNormGenerator (
    double meanlog_ = 0.0,
    double sdlog_ = 1.0 ) [inline]
```

Definition at line 31 of file rlnorm.h.

### 6.358.3 Member Function Documentation

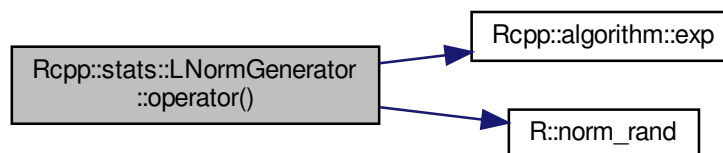
#### 6.358.3.1 operator()()

```
double Rcpp::stats::LNormGenerator::operator() ( ) const [inline]
```

Definition at line 34 of file rlnorm.h.

References [Rcpp::algorithm::exp\(\)](#), [meanlog](#), [R::norm\\_rand\(\)](#), and [sdlog](#).

Here is the call graph for this function:



## 6.358.4 Member Data Documentation

### 6.358.4.1 meanlog

```
double Rcpp::stats::LNormGenerator::meanlog [private]
```

Definition at line 39 of file rlnorm.h.

Referenced by operator(()).

### 6.358.4.2 sdlog

```
double Rcpp::stats::LNormGenerator::sdlog [private]
```

Definition at line 40 of file rlnorm.h.

Referenced by operator(()).

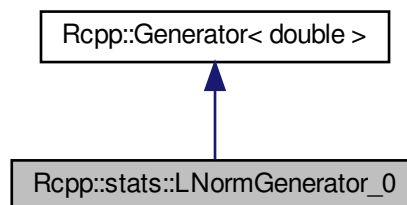
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rlnorm.h](#)

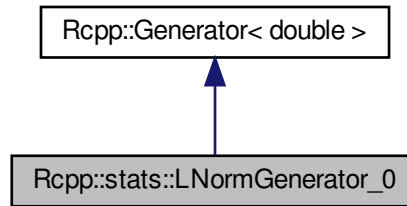
## 6.359 Rcpp::stats::LNormGenerator\_0 Class Reference

```
#include <rlnorm.h>
```

Inheritance diagram for Rcpp::stats::LNormGenerator\_0:



Collaboration diagram for Rcpp::stats::LNormGenerator\_0:



## Public Member Functions

- [LNormGenerator\\_0\(\)](#)
- double [operator\(\)](#) () const

## Additional Inherited Members

### 6.359.1 Detailed Description

Definition at line 59 of file rlnorm.h.

### 6.359.2 Constructor & Destructor Documentation

#### 6.359.2.1 LNormGenerator\_0()

```
Rcpp::stats::LNormGenerator_0::LNormGenerator_0 ( ) [inline]
```

Definition at line 62 of file rlnorm.h.

### 6.359.3 Member Function Documentation

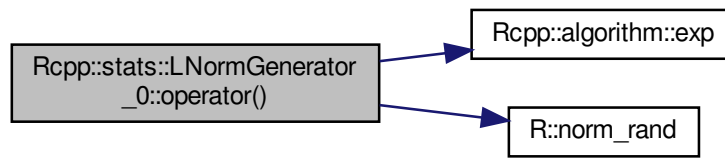
## 6.359.3.1 operator()

```
double Rcpp::stats::LNormGenerator_0::operator() ( ) const [inline]
```

Definition at line 64 of file rlnorm.h.

References `Rcpp::algorithm::exp()`, and `R::norm_rand()`.

Here is the call graph for this function:



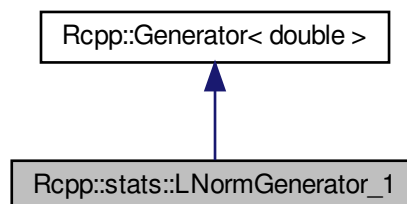
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rlnorm.h](#)

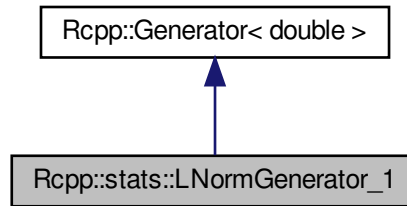
## 6.360 Rcpp::stats::LNormGenerator\_1 Class Reference

```
#include <rlnorm.h>
```

Inheritance diagram for `Rcpp::stats::LNormGenerator_1`:



Collaboration diagram for Rcpp::stats::LNormGenerator\_1:



## Public Member Functions

- [LNormGenerator\\_1](#) (double meanlog\_=0.0)
- double [operator\(\)](#) () const

## Private Attributes

- double [meanlog](#)

## Additional Inherited Members

### 6.360.1 Detailed Description

Definition at line 44 of file rlnorm.h.

### 6.360.2 Constructor & Destructor Documentation

#### 6.360.2.1 LNormGenerator\_1()

```
Rcpp::stats::LNormGenerator_1::LNormGenerator_1 (  
    double meanlog_ = 0.0 ) [inline]
```

Definition at line 47 of file rlnorm.h.



### 6.360.3 Member Function Documentation

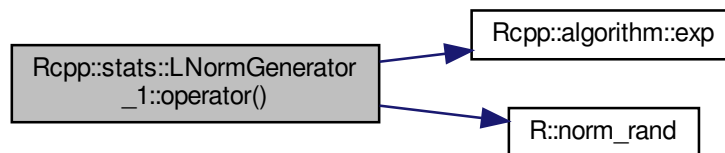
#### 6.360.3.1 operator()

```
double Rcpp::stats::LNormGenerator_1::operator() ( ) const [inline]
```

Definition at line 50 of file rlnorm.h.

References `Rcpp::algorithm::exp()`, `meanlog`, and `R::norm_rand()`.

Here is the call graph for this function:



### 6.360.4 Member Data Documentation

#### 6.360.4.1 meanlog

```
double Rcpp::stats::LNormGenerator_1::meanlog [private]
```

Definition at line 55 of file rlnorm.h.

Referenced by `operator()`.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rlnorm.h](#)

## 6.361 Rcpp::algorithm::helpers::log Struct Reference

```
#include <algorithm.h>
```

## Public Member Functions

- `template<typename T >`  
`double operator() (T val)`

### 6.361.1 Detailed Description

Definition at line 195 of file `algorithm.h`.

### 6.361.2 Member Function Documentation

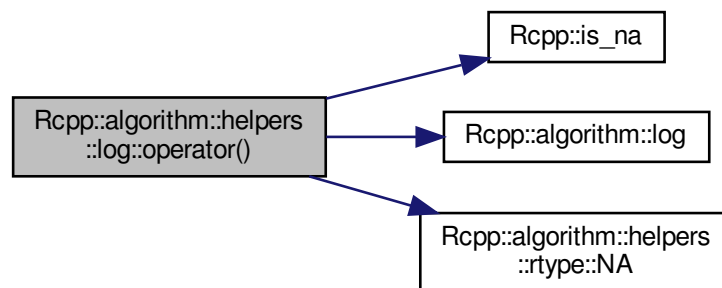
#### 6.361.2.1 operator>()

```
template<typename T >
double Rcpp::algorithm::helpers::log::operator() (
    T val ) [inline]
```

Definition at line 197 of file `algorithm.h`.

References `Rcpp::is_na()`, `Rcpp::algorithm::log()`, and `Rcpp::algorithm::helpers::rtype< T >::NA()`.

Here is the call graph for this function:



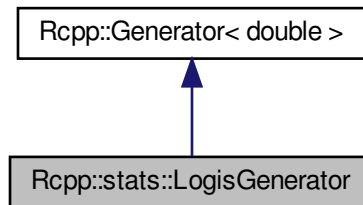
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/algorithm.h`

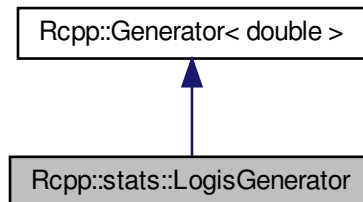
## 6.362 Rcpp::stats::LogisGenerator Class Reference

```
#include <rlogis.h>
```

Inheritance diagram for Rcpp::stats::LogisGenerator:



Collaboration diagram for Rcpp::stats::LogisGenerator:



### Public Member Functions

- [LogisGenerator](#) (double location\_, double scale\_)
- double [operator\(\)](#) () const

### Private Attributes

- double [location](#)
- double [scale](#)

## Additional Inherited Members

### 6.362.1 Detailed Description

Definition at line 28 of file rlogis.h.

### 6.362.2 Constructor & Destructor Documentation

#### 6.362.2.1 LogisGenerator()

```
Rcpp::stats::LogisGenerator::LogisGenerator (
    double location_,
    double scale_ ) [inline]
```

Definition at line 31 of file rlogis.h.

### 6.362.3 Member Function Documentation

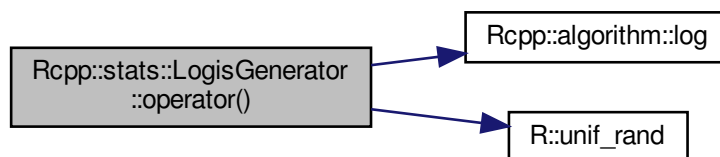
#### 6.362.3.1 operator()()

```
double Rcpp::stats::LogisGenerator::operator() ( ) const [inline]
```

Definition at line 34 of file rlogis.h.

References [location](#), [Rcpp::algorithm::log\(\)](#), [scale](#), and [R::unif\\_rand\(\)](#).

Here is the call graph for this function:



## 6.362.4 Member Data Documentation

### 6.362.4.1 location

```
double Rcpp::stats::LogisGenerator::location [private]
```

Definition at line 40 of file rlogis.h.

Referenced by operator()).

### 6.362.4.2 scale

```
double Rcpp::stats::LogisGenerator::scale [private]
```

Definition at line 41 of file rlogis.h.

Referenced by operator()).

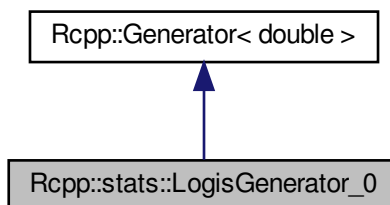
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rlogis.h](#)

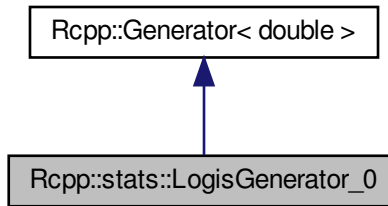
## 6.363 Rcpp::stats::LogisGenerator\_0 Class Reference

```
#include <rlogis.h>
```

Inheritance diagram for Rcpp::stats::LogisGenerator\_0:



Collaboration diagram for Rcpp::stats::LogisGenerator\_0:



## Public Member Functions

- [LogisGenerator\\_0\(\)](#)
- `double operator()() const`

## Additional Inherited Members

### 6.363.1 Detailed Description

Definition at line 61 of file `rlogis.h`.

### 6.363.2 Constructor & Destructor Documentation

#### 6.363.2.1 LogisGenerator\_0()

```
Rcpp::stats::LogisGenerator_0::LogisGenerator_0 ( ) [inline]
```

Definition at line 64 of file `rlogis.h`.

### 6.363.3 Member Function Documentation

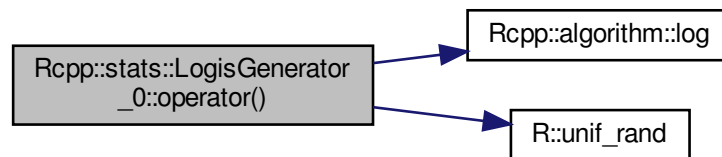
### 6.363.3.1 operator()

```
double Rcpp::stats::LogisGenerator_0::operator() ( ) const [inline]
```

Definition at line 66 of file rlogis.h.

References `Rcpp::algorithm::log()`, and `R::unif_rand()`.

Here is the call graph for this function:



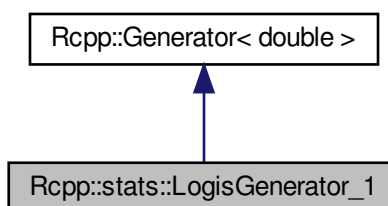
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rlogis.h](#)

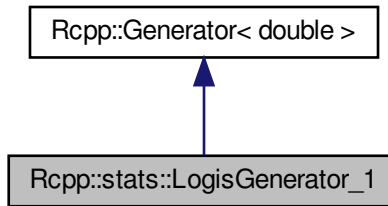
## 6.364 Rcpp::stats::LogisGenerator\_1 Class Reference

```
#include <rlogis.h>
```

Inheritance diagram for `Rcpp::stats::LogisGenerator_1`:



Collaboration diagram for Rcpp::stats::LogisGenerator\_1:



## Public Member Functions

- [LogisGenerator\\_1](#) (double location\_)
- double [operator\(\)](#) () const

## Private Attributes

- double [location](#)

## Additional Inherited Members

### 6.364.1 Detailed Description

Definition at line 45 of file rlogis.h.

### 6.364.2 Constructor & Destructor Documentation

#### 6.364.2.1 LogisGenerator\_1()

```
Rcpp::stats::LogisGenerator_1::LogisGenerator_1 (  
    double location_ ) [inline]
```

Definition at line 48 of file rlogis.h.



## 6.364.3 Member Function Documentation

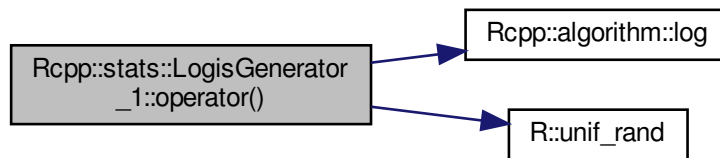
### 6.364.3.1 operator()()

```
double Rcpp::stats::LogisGenerator_1::operator() ( ) const [inline]
```

Definition at line 51 of file rlogis.h.

References `location`, `Rcpp::algorithm::log()`, and `R::unif_rand()`.

Here is the call graph for this function:



## 6.364.4 Member Data Documentation

### 6.364.4.1 location

```
double Rcpp::stats::LogisGenerator_1::location [private]
```

Definition at line 57 of file rlogis.h.

Referenced by `operator()()`.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rlogis.h](#)

## 6.365 Rcpp::LongjumpException Struct Reference

```
#include <exceptions.h>
```

## Public Member Functions

- [LongjumpException](#) (SEXP token\_)

## Public Attributes

- SEXP [token](#)

### 6.365.1 Detailed Description

Definition at line 164 of file exceptions.h.

### 6.365.2 Constructor & Destructor Documentation

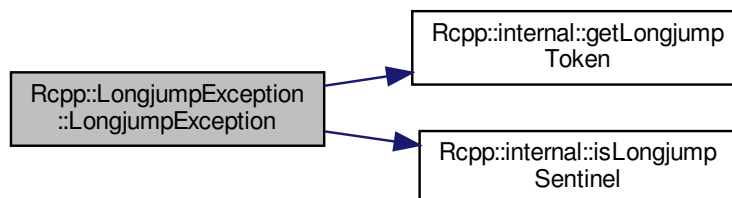
#### 6.365.2.1 LongjumpException()

```
Rcpp::LongjumpException::LongjumpException (  
    SEXP token_ ) [inline]
```

Definition at line 166 of file exceptions.h.

References [Rcpp::internal::getLongjumpToken\(\)](#), [Rcpp::internal::isLongjumpSentinel\(\)](#), and [token](#).

Here is the call graph for this function:



### 6.365.3 Member Data Documentation

### 6.365.3.1 token

```
SEXP Rcpp::LongjumpException::token
```

Definition at line 165 of file exceptions.h.

Referenced by LongjumpException().

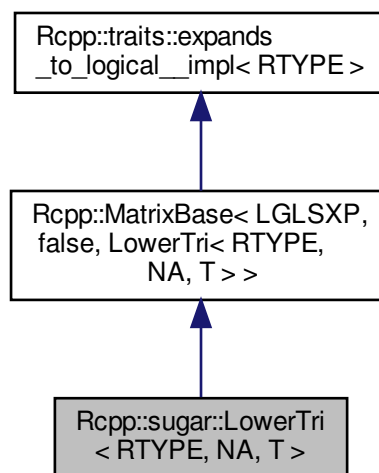
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/exceptions.h](#)

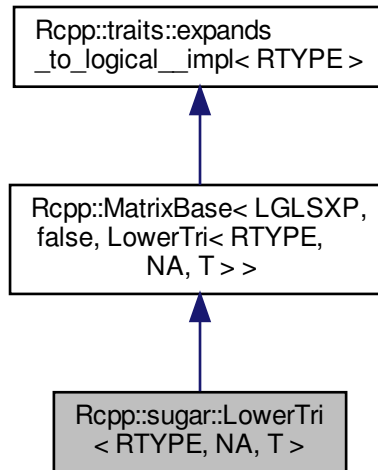
## 6.366 Rcpp::sugar::LowerTri< RTYPE, NA, T > Class Template Reference

```
#include <lower_tri.h>
```

Inheritance diagram for Rcpp::sugar::LowerTri< RTYPE, NA, T >:



Collaboration diagram for `Rcpp::sugar::LowerTri< RTYPE, NA, T >`:



## Public Types

- typedef `Rcpp::MatrixBase< RTYPE, NA, T >` `MatBase`

## Public Member Functions

- `LowerTri` (const T &lhs, bool `diag`)
- int `operator()` (int i, int j) const
- `R_xlen_t size` () const
- int `nrow` () const
- int `ncol` () const

## Private Types

- typedef bool(LowerTri::\* `Method`) (int, int) const

## Private Member Functions

- bool `get_diag_true` (int i, int j) const
- bool `get_diag_false` (int i, int j) const
- bool `get` (int i, int j) const

## Private Attributes

- int `nr`
- int `nc`
- Method `getter`

## Additional Inherited Members

### 6.366.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::LowerTri< RTYPE, NA, T >
```

Definition at line 30 of file `lower_tri.h`.

### 6.366.2 Member Typedef Documentation

#### 6.366.2.1 MatBase

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::MatrixBase<RTYPE, NA, T> Rcpp::sugar::LowerTri< RTYPE, NA, T >::MatBase
```

Definition at line 32 of file `lower_tri.h`.

#### 6.366.2.2 Method

```
template<int RTYPE, bool NA, typename T >  
typedef bool(LowerTri::* Rcpp::sugar::LowerTri< RTYPE, NA, T >::Method) (int, int) const [private]
```

Definition at line 47 of file `lower_tri.h`.

### 6.366.3 Constructor & Destructor Documentation

### 6.366.3.1 LowerTri()

```
template<int RTYPE, bool NA, typename T >  
Rcpp::sugar::LowerTri< RTYPE, NA, T >::LowerTri (  
    const T & lhs,  
    bool diag ) [inline]
```

Definition at line 34 of file lower\_tri.h.

## 6.366.4 Member Function Documentation

### 6.366.4.1 get()

```
template<int RTYPE, bool NA, typename T >  
bool Rcpp::sugar::LowerTri< RTYPE, NA, T >::get (  
    int i,  
    int j ) const [inline], [private]
```

Definition at line 56 of file lower\_tri.h.

Referenced by Rcpp::sugar::LowerTri< RTYPE, NA, T >::operator()().

### 6.366.4.2 get\_diag\_false()

```
template<int RTYPE, bool NA, typename T >  
bool Rcpp::sugar::LowerTri< RTYPE, NA, T >::get_diag_false (  
    int i,  
    int j ) const [inline], [private]
```

Definition at line 54 of file lower\_tri.h.

### 6.366.4.3 get\_diag\_true()

```
template<int RTYPE, bool NA, typename T >  
bool Rcpp::sugar::LowerTri< RTYPE, NA, T >::get_diag_true (  
    int i,  
    int j ) const [inline], [private]
```

Definition at line 52 of file lower\_tri.h.

#### 6.366.4.4 ncol()

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::sugar::LowerTri< RTYPE, NA, T >::ncol ( ) const [inline]
```

Definition at line 44 of file lower\_tri.h.

References Rcpp::sugar::LowerTri< RTYPE, NA, T >::nc.

#### 6.366.4.5 nrow()

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::sugar::LowerTri< RTYPE, NA, T >::nrow ( ) const [inline]
```

Definition at line 43 of file lower\_tri.h.

References Rcpp::sugar::LowerTri< RTYPE, NA, T >::nr.

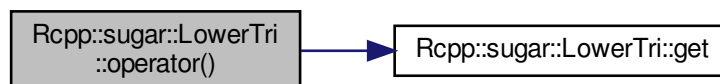
#### 6.366.4.6 operator()()

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::sugar::LowerTri< RTYPE, NA, T >::operator() (   
    int i,   
    int j ) const [inline]
```

Definition at line 40 of file lower\_tri.h.

References Rcpp::sugar::LowerTri< RTYPE, NA, T >::get().

Here is the call graph for this function:



### 6.366.4.7 size()

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::LowerTri< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 42 of file lower\_tri.h.

References Rcpp::sugar::LowerTri< RTYPE, NA, T >::nc, and Rcpp::sugar::LowerTri< RTYPE, NA, T >::nr.

## 6.366.5 Member Data Documentation

### 6.366.5.1 getter

```
template<int RTYPE, bool NA, typename T >  
Method Rcpp::sugar::LowerTri< RTYPE, NA, T >::getter [private]
```

Definition at line 50 of file lower\_tri.h.

### 6.366.5.2 nc

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::sugar::LowerTri< RTYPE, NA, T >::nc [private]
```

Definition at line 49 of file lower\_tri.h.

Referenced by Rcpp::sugar::LowerTri< RTYPE, NA, T >::ncol(), and Rcpp::sugar::LowerTri< RTYPE, NA, T >::size().

### 6.366.5.3 nr

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::sugar::LowerTri< RTYPE, NA, T >::nr [private]
```

Definition at line 49 of file lower\_tri.h.

Referenced by Rcpp::sugar::LowerTri< RTYPE, NA, T >::nrow(), and Rcpp::sugar::LowerTri< RTYPE, NA, T >::size().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/lower\_tri.h



## 6.367 Rcpp::lsinfo Struct Reference

### Public Attributes

- [time\\_t ls\\_trans](#)
- [int\\_fast64\\_t ls\\_corr](#)

### 6.367.1 Detailed Description

Definition at line 388 of file date.cpp.

### 6.367.2 Member Data Documentation

#### 6.367.2.1 ls\_corr

```
int_fast64_t Rcpp::lsinfo::ls_corr
```

Definition at line 390 of file date.cpp.

Referenced by [Rcpp::timesub\(\)](#), and [Rcpp::tzload\(\)](#).

#### 6.367.2.2 ls\_trans

```
time_t Rcpp::lsinfo::ls_trans
```

Definition at line 389 of file date.cpp.

Referenced by [Rcpp::timesub\(\)](#), and [Rcpp::tzload\(\)](#).

The documentation for this struct was generated from the following file:

- [src/date.cpp](#)

## 6.368 Rcpp::sugar::cbind\_impl::detail::MakeBindableCall< T, is\_container > Struct Template Reference

```
#include <cbind.h>
```

### 6.368.1 Detailed Description

```
template<typename T, bool is_container = has_stored_type<T>::value>
struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, is_container >
```

Definition at line 392 of file cbind.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/cbind.h](#)

## 6.369 Rcpp::sugar::cbind\_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true > Struct Reference

```
#include <cbind.h>
```

### Public Types

- enum { [RTYPE](#) = cbind\_sexptype\_traits<stored\_type>::rtype }
- typedef [Rcpp::Matrix< LGLSXP > T](#)
- typedef bool [stored\\_type](#)

### Public Member Functions

- [ContainerBindable< LGLSXP, T > operator\(\)](#) (const T &t) const

### 6.369.1 Detailed Description

Definition at line 409 of file cbind.h.

### 6.369.2 Member Typedef Documentation

#### 6.369.2.1 stored\_type

```
typedef bool Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true >↵
::stored_type
```

Definition at line 411 of file cbind.h.

### 6.369.2.2 T

```
typedef Rcpp::Matrix<LGLSXP> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix<  
LGLSXP >, true >::T
```

Definition at line 410 of file cbind.h.

## 6.369.3 Member Enumeration Documentation

### 6.369.3.1 anonymous enum

anonymous enum

Enumerator

RTYPE
-------

Definition at line 413 of file cbind.h.

## 6.369.4 Member Function Documentation

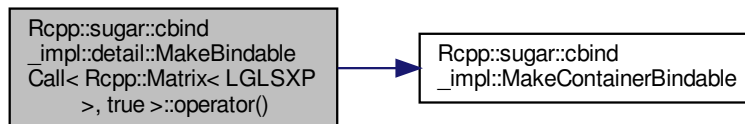
### 6.369.4.1 operator>()

```
ContainerBindable<LGLSXP, T> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Matrix<  
LGLSXP >, true >::operator() (  
    const T & t ) const [inline]
```

Definition at line 415 of file cbind.h.

References Rcpp::sugar::cbind\_impl::MakeContainerBindable().

Here is the call graph for this function:



The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/cbind.h](#)

## 6.370 Rcpp::sugar::cbind\_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true > Struct Reference

```
#include <cbind.h>
```

### Public Types

- enum { [RTYPE](#) = [cbind\\_sexptype\\_traits<stored\\_type>::rtype](#) }
- typedef [Rcpp::Vector< LGLSXP > T](#)
- typedef bool [stored\\_type](#)

### Public Member Functions

- [ContainerBindable< LGLSXP, T > operator\(\)](#) (const T &t) const

#### 6.370.1 Detailed Description

Definition at line 421 of file [cbind.h](#).

#### 6.370.2 Member Typedef Documentation

##### 6.370.2.1 stored\_type

```
typedef bool Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true >↵  
::stored_type
```

Definition at line 423 of file [cbind.h](#).

##### 6.370.2.2 T

```
typedef Rcpp::Vector<LGLSXP> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector<  
LGLSXP >, true >::T
```

Definition at line 422 of file [cbind.h](#).

#### 6.370.3 Member Enumeration Documentation

##### 6.370.3.1 anonymous enum

```
anonymous enum
```

Enumerator

RTYPE	
-------	--

Definition at line 425 of file cbind.h.

## 6.370.4 Member Function Documentation

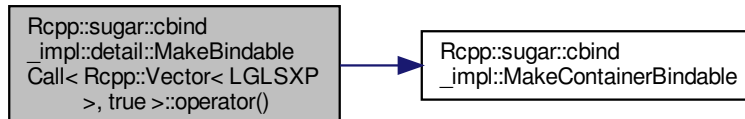
### 6.370.4.1 operator>()

```
ContainerBindable<LGLSXP, T> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< Rcpp::Vector<
LGLSXP >, true >::operator() (
    const T & t ) const [inline]
```

Definition at line 427 of file cbind.h.

References Rcpp::sugar::cbind\_impl::MakeContainerBindable().

Here is the call graph for this function:



The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/cbind.h](#)

## 6.371 Rcpp::sugar::cbind\_impl::detail::MakeBindableCall< T, false > Struct Template Reference

```
#include <cbind.h>
```

## Public Types

- enum { `RTYPE` = `cbind_sexptype_traits<T>::rtype` }

## Public Member Functions

- `ScalarBindable< T > operator()` (const T &t) const

### 6.371.1 Detailed Description

```
template<typename T>
struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, false >
```

Definition at line 433 of file `cbind.h`.

### 6.371.2 Member Enumeration Documentation

#### 6.371.2.1 anonymous enum

```
template<typename T >
anonymous enum
```

Enumerator

RTYPE	
-------	--

Definition at line 434 of file `cbind.h`.

### 6.371.3 Member Function Documentation

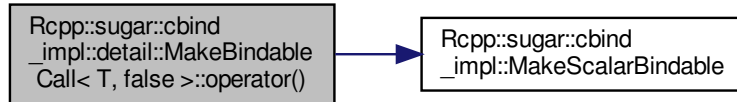
#### 6.371.3.1 operator>()

```
template<typename T >
ScalarBindable<T> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, false >::operator() (
    const T & t ) const [inline]
```

Definition at line 436 of file `cbind.h`.

References Rcpp::sugar::cbind\_impl::MakeScalarBindable().

Here is the call graph for this function:



The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/cbind.h](#)

## 6.372 Rcpp::sugar::cbind\_impl::detail::MakeBindableCall< T, true > Struct Template Reference

```
#include <cbind.h>
```

### Public Types

- enum { [RTYPE](#) = cbind\_sexptype\_traits<stored\_type>::rtype }
- typedef [cbind\\_storage\\_type](#)< [cbind\\_sexptype\\_traits](#)< typename T::stored\_type >::rtype >::type [stored\\_type](#)

### Public Member Functions

- [ContainerBindable](#)< [RTYPE](#), T > [operator\(\)](#) (const T &t) const

#### 6.372.1 Detailed Description

```
template<typename T>
struct Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, true >
```

Definition at line 395 of file cbind.h.

#### 6.372.2 Member Typedef Documentation

### 6.372.2.1 stored\_type

```
template<typename T >
typedef cbind_storage_type< cbind_sexptype_traits<typename T::stored_type>::rtype >::type Rcpp::sugar::cbind_imp
T, true >::stored_type
```

Definition at line 398 of file cbind.h.

## 6.372.3 Member Enumeration Documentation

### 6.372.3.1 anonymous enum

```
template<typename T >
anonymous enum
```

Enumerator

RTYPE	
-------	--

Definition at line 400 of file cbind.h.

## 6.372.4 Member Function Documentation

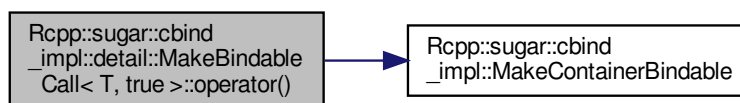
### 6.372.4.1 operator>()

```
template<typename T >
ContainerBindable<RTYPE, T> Rcpp::sugar::cbind_impl::detail::MakeBindableCall< T, true >::operator()
(
    const T & t ) const [inline]
```

Definition at line 402 of file cbind.h.

References Rcpp::sugar::cbind\_impl::MakeContainerBindable().

Here is the call graph for this function:





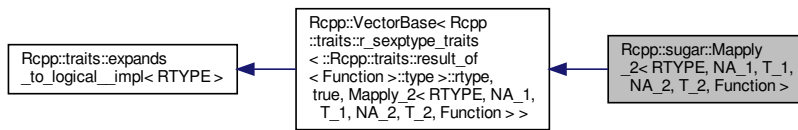
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

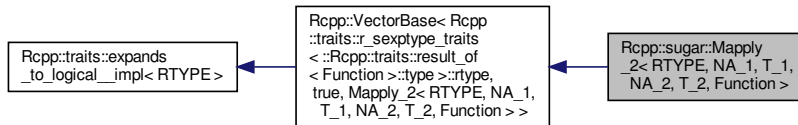
## 6.373 Rcpp::sugar::Mapply\_2< RTYPE, NA\_1, T\_1, NA\_2, T\_2, Function > Class Template Reference

```
#include <mapply_2.h>
```

Inheritance diagram for Rcpp::sugar::Mapply\_2< RTYPE, NA\_1, T\_1, NA\_2, T\_2, Function >:



Collaboration diagram for Rcpp::sugar::Mapply\_2< RTYPE, NA\_1, T\_1, NA\_2, T\_2, Function >:



### Public Types

- typedef `::Rcpp::traits::result_of< Function >::type` `result_type`

### Public Member Functions

- `Mapply_2` (const T\_1 &vec\_1\_, const T\_2 &vec\_2\_, Function fun\_)
- `result_type operator[]` (R\_xlen\_t i) const
- `R_xlen_t size` () const

### Private Attributes

- const T\_1 & `vec_1`
- const T\_2 & `vec_2`
- Function `fun`

### 6.373.1 Detailed Description

```
template<int RTYPE, bool NA_1, typename T_1, bool NA_2, typename T_2, typename Function>  
class Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >
```

Definition at line 33 of file mapply\_2.h.

### 6.373.2 Member Typedef Documentation

#### 6.373.2.1 result\_type

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >  
typedef ::Rcpp::traits::result_of<Function>::type Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2,  
T_2, Function >::result_type
```

Definition at line 41 of file mapply\_2.h.

### 6.373.3 Constructor & Destructor Documentation

#### 6.373.3.1 Mapply\_2()

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >  
Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::Mapply_2 (  
    const T_1 & vec_1_,  
    const T_2 & vec_2_,  
    Function fun_ ) [inline]
```

Definition at line 43 of file mapply\_2.h.

### 6.373.4 Member Function Documentation

### 6.373.4.1 operator[]()

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
result_type Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 46 of file mapply\_2.h.

References Rcpp::sugar::Mapply\_2< RTYPE, NA\_1, T\_1, NA\_2, T\_2, Function >::fun, Rcpp::sugar::Mapply\_2< RTYPE, NA\_1, T\_1, NA\_2, T\_2, Function >::vec\_1, and Rcpp::sugar::Mapply\_2< RTYPE, NA\_1, T\_1, NA\_2, T\_2, Function >::vec\_2.

### 6.373.4.2 size()

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
R_xlen_t Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::size ( ) const [inline]
```

Definition at line 49 of file mapply\_2.h.

References Rcpp::sugar::Mapply\_2< RTYPE, NA\_1, T\_1, NA\_2, T\_2, Function >::vec\_1.

## 6.373.5 Member Data Documentation

### 6.373.5.1 fun

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
Function Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::fun [private]
```

Definition at line 54 of file mapply\_2.h.

Referenced by Rcpp::sugar::Mapply\_2< RTYPE, NA\_1, T\_1, NA\_2, T\_2, Function >::operator[]().

### 6.373.5.2 vec\_1

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
const T_1& Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::vec_1 [private]
```

Definition at line 52 of file mapply\_2.h.

Referenced by Rcpp::sugar::Mapply\_2< RTYPE, NA\_1, T\_1, NA\_2, T\_2, Function >::operator[](), and Rcpp::sugar::Mapply\_2< RTYPE, NA\_1, T\_1, NA\_2, T\_2, Function >::size().

### 6.373.5.3 vec\_2

```
template<int RTYPE, bool NA_1, typename T_1 , bool NA_2, typename T_2 , typename Function >
const T_2& Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::vec_2 [private]
```

Definition at line 53 of file mapply\_2.h.

Referenced by `Rcpp::sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >::operator[]()`.

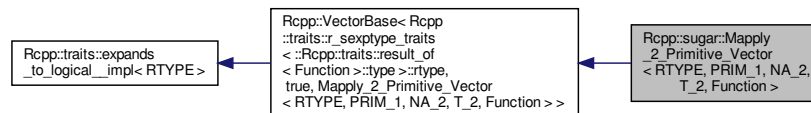
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/mapply/mapply\\_2.h](#)

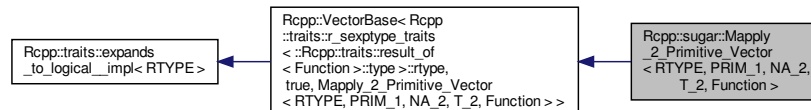
## 6.374 Rcpp::sugar::Mapply\_2\_Primitive\_Vector< RTYPE, PRIM\_1, NA\_2, T\_2, Function > Class Template Reference

```
#include <mapply_2.h>
```

Inheritance diagram for `Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >`:



Collaboration diagram for `Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >`:



## Public Types

- `typedef ::Rcpp::traits::result_of< Function >::type result_type`

## Public Member Functions

- [Mapply\\_2\\_Primitive\\_Vector](#) (PRIM\_1 prim\_1\_, const T\_2 &vec\_2\_, [Function](#) fun\_)
- [result\\_type operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- PRIM\_1 [prim\\_1](#)
- const T\_2 & [vec\\_2](#)
- [Function](#) [fun](#)

### 6.374.1 Detailed Description

```
template<int RTYPE, typename PRIM_1, bool NA_2, typename T_2, typename Function>
class Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >
```

Definition at line 93 of file `mapply_2.h`.

### 6.374.2 Member Typedef Documentation

#### 6.374.2.1 result\_type

```
template<int RTYPE, typename PRIM_1, bool NA_2, typename T_2, typename Function >
typedef ::Rcpp::traits::result_of<Function>::type Rcpp::sugar::Mapply\_2\_Primitive\_Vector< RTYPE,
PRIM_1, NA_2, T_2, Function >::result\_type
```

Definition at line 103 of file `mapply_2.h`.

### 6.374.3 Constructor & Destructor Documentation

#### 6.374.3.1 Mapply\_2\_Primitive\_Vector()

```
template<int RTYPE, typename PRIM_1, bool NA_2, typename T_2, typename Function >
Rcpp::sugar::Mapply\_2\_Primitive\_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::Mapply\_2\_Primitive\_Vector
(
    PRIM_1 prim\_1\_,
    const T_2 & vec\_2\_,
    Function fun\_ ) [inline]
```

Definition at line 105 of file `mapply_2.h`.

## 6.374.4 Member Function Documentation

### 6.374.4.1 operator[]()

```
template<int RTYPE, typename PRIM_1 , bool NA_2, typename T_2 , typename Function >
result_type Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::operator[]
(
    R_xlen_t i ) const [inline]
```

Definition at line 108 of file mapply\_2.h.

References [Rcpp::sugar::Mapply\\_2\\_Primitive\\_Vector< RTYPE, PRIM\\_1, NA\\_2, T\\_2, Function >::fun](#), [Rcpp::sugar::Mapply\\_2\\_Primitive\\_Vector< RTYPE, PRIM\\_1, NA\\_2, T\\_2, Function >::prim\\_1](#), and [Rcpp::sugar::Mapply\\_2\\_Primitive\\_Vector< RTYPE, PRIM\\_1, NA\\_2, T\\_2, Function >::vec\\_2](#).

### 6.374.4.2 size()

```
template<int RTYPE, typename PRIM_1 , bool NA_2, typename T_2 , typename Function >
R_xlen_t Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::size ( )
const [inline]
```

Definition at line 111 of file mapply\_2.h.

References [Rcpp::sugar::Mapply\\_2\\_Primitive\\_Vector< RTYPE, PRIM\\_1, NA\\_2, T\\_2, Function >::vec\\_2](#).

## 6.374.5 Member Data Documentation

### 6.374.5.1 fun

```
template<int RTYPE, typename PRIM_1 , bool NA_2, typename T_2 , typename Function >
Function Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::fun [private]
```

Definition at line 116 of file mapply\_2.h.

Referenced by [Rcpp::sugar::Mapply\\_2\\_Primitive\\_Vector< RTYPE, PRIM\\_1, NA\\_2, T\\_2, Function >::operator\[\]\(\)](#).

6.374.5.2 prim\_1

```
template<int RTYPE, typename PRIM_1 , bool NA_2, typename T_2 , typename Function >
PRIM_1 Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::prim_←
1 [private]
```

Definition at line 114 of file mapply\_2.h.

Referenced by Rcpp::sugar::Mapply\_2\_Primitive\_Vector< RTYPE, PRIM\_1, NA\_2, T\_2, Function >::operator[]().

6.374.5.3 vec\_2

```
template<int RTYPE, typename PRIM_1 , bool NA_2, typename T_2 , typename Function >
const T_2& Rcpp::sugar::Mapply_2_Primitive_Vector< RTYPE, PRIM_1, NA_2, T_2, Function >::vec_←
2 [private]
```

Definition at line 115 of file mapply\_2.h.

Referenced by Rcpp::sugar::Mapply\_2\_Primitive\_Vector< RTYPE, PRIM\_1, NA\_2, T\_2, Function >::operator[](), and Rcpp::sugar::Mapply\_2\_Primitive\_Vector< RTYPE, PRIM\_1, NA\_2, T\_2, Function >::size().

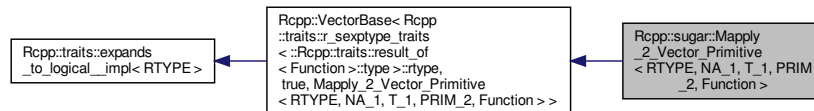
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mapply/mapply\_2.h

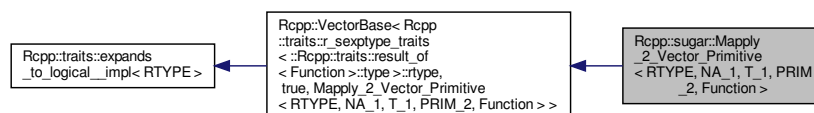
6.375 Rcpp::sugar::Mapply\_2\_Vector\_Primitive< RTYPE, NA\_1, T\_1, PRIM\_2, Function > Class Template Reference

```
#include <mapply_2.h>
```

Inheritance diagram for Rcpp::sugar::Mapply\_2\_Vector\_Primitive< RTYPE, NA\_1, T\_1, PRIM\_2, Function >:



Collaboration diagram for Rcpp::sugar::Mapply\_2\_Vector\_Primitive< RTYPE, NA\_1, T\_1, PRIM\_2, Function >:



## Public Types

- `typedef ::Rcpp::traits::result_of< Function >::type result_type`

## Public Member Functions

- `Mapply_2_Vector_Primitive` (const T\_1 &vec\_1\_, PRIM\_2 prim\_2\_, Function fun\_)
- `result_type operator[]` (R\_xlen\_t i) const
- `R_xlen_t size` () const

## Private Attributes

- const T\_1 & `vec_1`
- PRIM\_2 `prim_2`
- Function `fun`

### 6.375.1 Detailed Description

```
template<int RTYPE, bool NA_1, typename T_1, typename PRIM_2, typename Function>
class Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >
```

Definition at line 62 of file `mapply_2.h`.

### 6.375.2 Member Typedef Documentation

#### 6.375.2.1 result\_type

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >
typedef ::Rcpp::traits::result_of<Function>::type Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE,
NA_1, T_1, PRIM_2, Function >::result_type
```

Definition at line 72 of file `mapply_2.h`.

### 6.375.3 Constructor & Destructor Documentation



### 6.375.3.1 Mapply\_2\_Vector\_Primitive()

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >  
Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::Mapply_2_Vector_Primitive  
(  
    const T_1 & vec_1_,  
    PRIM_2 prim_2_,  
    Function fun_ ) [inline]
```

Definition at line 74 of file mapply\_2.h.

## 6.375.4 Member Function Documentation

### 6.375.4.1 operator[]()

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >  
result_type Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::operator[]  
(  
    R_xlen_t i ) const [inline]
```

Definition at line 77 of file mapply\_2.h.

References Rcpp::sugar::Mapply\_2\_Vector\_Primitive< RTYPE, NA\_1, T\_1, PRIM\_2, Function >::fun, Rcpp::sugar::Mapply\_2\_Vector\_Primitive< RTYPE, NA\_1, T\_1, PRIM\_2, Function >::prim\_2, and Rcpp::sugar::Mapply\_2\_Vector\_Primitive< RTYPE, NA\_1, T\_1, PRIM\_2, Function >::vec\_1.

### 6.375.4.2 size()

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >  
R_xlen_t Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::size ( )  
const [inline]
```

Definition at line 80 of file mapply\_2.h.

References Rcpp::sugar::Mapply\_2\_Vector\_Primitive< RTYPE, NA\_1, T\_1, PRIM\_2, Function >::vec\_1.

## 6.375.5 Member Data Documentation

### 6.375.5.1 fun

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >  
Function Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::fun [private]
```

Definition at line 85 of file mapply\_2.h.

Referenced by Rcpp::sugar::Mapply\_2\_Vector\_Primitive< RTYPE, NA\_1, T\_1, PRIM\_2, Function >::operator[ ]().

### 6.375.5.2 prim\_2

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >  
PRIM_2 Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::prim_↔  
2 [private]
```

Definition at line 84 of file mapply\_2.h.

Referenced by Rcpp::sugar::Mapply\_2\_Vector\_Primitive< RTYPE, NA\_1, T\_1, PRIM\_2, Function >::operator[ ]().

### 6.375.5.3 vec\_1

```
template<int RTYPE, bool NA_1, typename T_1 , typename PRIM_2 , typename Function >  
const T_1& Rcpp::sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, PRIM_2, Function >::vec_↔  
1 [private]
```

Definition at line 83 of file mapply\_2.h.

Referenced by Rcpp::sugar::Mapply\_2\_Vector\_Primitive< RTYPE, NA\_1, T\_1, PRIM\_2, Function >::operator[ ](), and Rcpp::sugar::Mapply\_2\_Vector\_Primitive< RTYPE, NA\_1, T\_1, PRIM\_2, Function >::size().

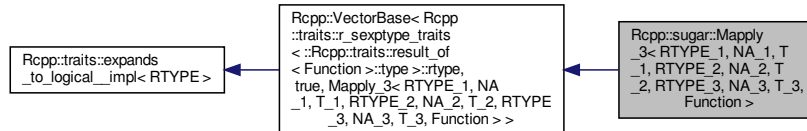
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/mapply/mapply\\_2.h](#)

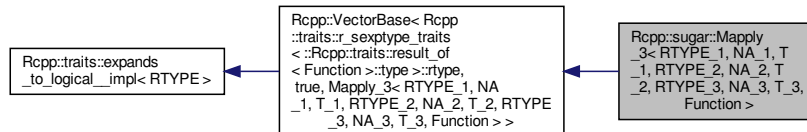
## 6.376 Rcpp::sugar::Mapply\_3< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function > Class Template Reference

```
#include <mapply_3.h>
```

Inheritance diagram for Rcpp::sugar::Mapply\_3< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function >:



Collaboration diagram for Rcpp::sugar::Mapply\_3< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function >:



### Public Types

- typedef `Rcpp::traits::result_of< Function >::type` `result_type`
- typedef `Rcpp::VectorBase< RTYPE_1, NA_1, T_1 >` `VEC_1`
- typedef `Rcpp::VectorBase< RTYPE_2, NA_2, T_2 >` `VEC_2`
- typedef `Rcpp::VectorBase< RTYPE_3, NA_3, T_3 >` `VEC_3`
- typedef `Rcpp::traits::Extractor< RTYPE_1, NA_1, T_1 >::type` `EXT_1`
- typedef `Rcpp::traits::Extractor< RTYPE_2, NA_2, T_2 >::type` `EXT_2`
- typedef `Rcpp::traits::Extractor< RTYPE_3, NA_3, T_3 >::type` `EXT_3`

### Public Member Functions

- `Mapply_3` (const `VEC_1` &vec\_1\_, const `VEC_2` &vec\_2\_, const `VEC_3` &vec\_3\_, `Function` fun\_)
- `result_type operator[]` (R\_xlen\_t i) const
- `R_xlen_t size` () const

## Private Attributes

- const [EXT\\_1](#) & [vec\\_1](#)
- const [EXT\\_2](#) & [vec\\_2](#)
- const [EXT\\_3](#) & [vec\\_3](#)
- [Function](#) fun

### 6.376.1 Detailed Description

```
template<int RTYPE_1, bool NA_1, typename T_1, int RTYPE_2, bool NA_2, typename T_2, int RTYPE_3, bool NA_3, typename T_3,
typename Function>
```

```
class Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >
```

Definition at line 34 of file `mapply_3.h`.

### 6.376.2 Member Typedef Documentation

#### 6.376.2.1 EXT\_1

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_←
_3, bool NA_3, typename T_3 , typename Function >
typedef Rcpp::traits::Extractor<RTYPE_1,NA_1,T_1>::type Rcpp::sugar::Mapply\_3< RTYPE_1, NA_1, T_←
_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::EXT_1
```

Definition at line 48 of file `mapply_3.h`.

#### 6.376.2.2 EXT\_2

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_←
_3, bool NA_3, typename T_3 , typename Function >
typedef Rcpp::traits::Extractor<RTYPE_2,NA_2,T_2>::type Rcpp::sugar::Mapply\_3< RTYPE_1, NA_1, T_←
_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::EXT_2
```

Definition at line 49 of file `mapply_3.h`.

### 6.376.2.3 EXT\_3

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_3, bool NA_3, typename T_3 , typename Function >  
typedef Rcpp::traits::Extractor<RTYPE_3,NA_3,T_3>::type Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::EXT_3
```

Definition at line 50 of file mapply\_3.h.

### 6.376.2.4 result\_type

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_3, bool NA_3, typename T_3 , typename Function >  
typedef ::Rcpp::traits::result_of<Function>::type Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::result_type
```

Definition at line 42 of file mapply\_3.h.

### 6.376.2.5 VEC\_1

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_3, bool NA_3, typename T_3 , typename Function >  
typedef Rcpp::VectorBase<RTYPE_1,NA_1,T_1> Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::VEC_1
```

Definition at line 44 of file mapply\_3.h.

### 6.376.2.6 VEC\_2

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_3, bool NA_3, typename T_3 , typename Function >  
typedef Rcpp::VectorBase<RTYPE_2,NA_2,T_2> Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::VEC_2
```

Definition at line 45 of file mapply\_3.h.

### 6.376.2.7 VEC\_3

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_3, bool NA_3, typename T_3 , typename Function >
typedef Rcpp::VectorBase<RTYPE_3,NA_3,T_3> Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::VEC_3
```

Definition at line 46 of file mapply\_3.h.

## 6.376.3 Constructor & Destructor Documentation

### 6.376.3.1 Mapply\_3()

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_3, bool NA_3, typename T_3 , typename Function >
Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::Mapply_3 (
    const VEC_1 & vec_1_,
    const VEC_2 & vec_2_,
    const VEC_3 & vec_3_,
    Function fun_ ) [inline]
```

Definition at line 52 of file mapply\_3.h.

## 6.376.4 Member Function Documentation

### 6.376.4.1 operator[]()

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_3, bool NA_3, typename T_3 , typename Function >
result_type Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 55 of file mapply\_3.h.

References Rcpp::sugar::Mapply\_3< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function >::fun, Rcpp::sugar::Mapply\_3< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function >::vec\_1, Rcpp::sugar::Mapply\_3< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function >::vec\_2, and Rcpp::sugar::Mapply\_3< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function >::vec\_3.

### 6.376.4.2 size()

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_3, bool NA_3, typename T_3 , typename Function >  
R_xlen_t Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::size ( ) const [inline]
```

Definition at line 58 of file mapply\_3.h.

References Rcpp::sugar::Mapply\_3< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function >::vec\_1.

## 6.376.5 Member Data Documentation

### 6.376.5.1 fun

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_3, bool NA_3, typename T_3 , typename Function >  
Function Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::fun [private]
```

Definition at line 64 of file mapply\_3.h.

Referenced by Rcpp::sugar::Mapply\_3< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function >::operator[ ]().

### 6.376.5.2 vec\_1

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_3, bool NA_3, typename T_3 , typename Function >  
const EXT_1& Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::vec_1 [private]
```

Definition at line 61 of file mapply\_3.h.

Referenced by Rcpp::sugar::Mapply\_3< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function >::operator[ ](), and Rcpp::sugar::Mapply\_3< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function >::size().

### 6.376.5.3 vec\_2

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_3, bool NA_3, typename T_3 , typename Function >
const EXT_2& Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::vec_2 [private]
```

Definition at line 62 of file mapply\_3.h.

Referenced by Rcpp::sugar::Mapply\_3< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function >::operator[]().

### 6.376.5.4 vec\_3

```
template<int RTYPE_1, bool NA_1, typename T_1 , int RTYPE_2, bool NA_2, typename T_2 , int RTYPE_3, bool NA_3, typename T_3 , typename Function >
const EXT_3& Rcpp::sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >::vec_3 [private]
```

Definition at line 63 of file mapply\_3.h.

Referenced by Rcpp::sugar::Mapply\_3< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function >::operator[]().

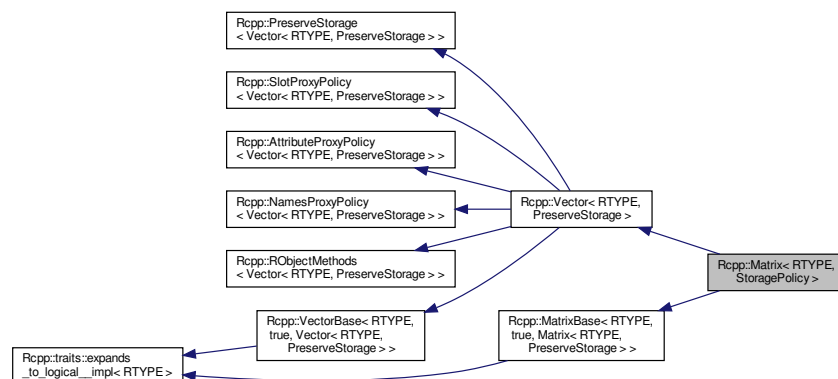
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/mapply/mapply\_3.h

## 6.377 Rcpp::Matrix< RTYPE, StoragePolicy > Class Template Reference

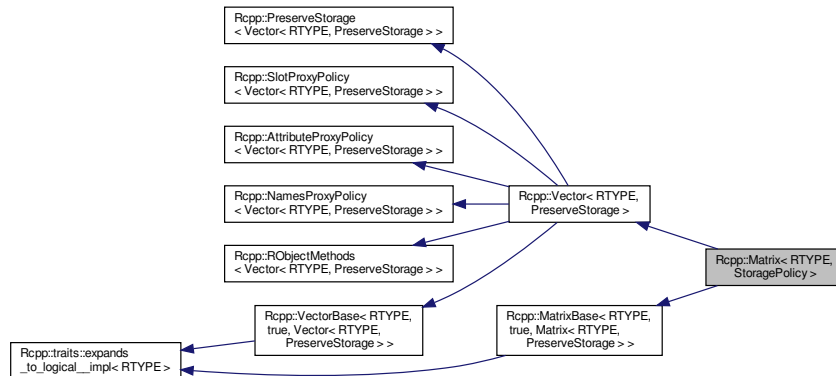
```
#include <Matrix.h>
```

Inheritance diagram for Rcpp::Matrix< RTYPE, StoragePolicy >:





Collaboration diagram for Rcpp::Matrix< RTYPE, StoragePolicy > :



## Classes

- struct [can\\_have\\_na](#)
- struct [r\\_type](#)

## Public Types

- typedef [MatrixRow](#)< RTYPE > [Row](#)
- typedef [ConstMatrixRow](#)< RTYPE > [ConstRow](#)
- typedef [MatrixColumn](#)< RTYPE > [Column](#)
- typedef [ConstMatrixColumn](#)< RTYPE > [ConstColumn](#)
- typedef [SubMatrix](#)< RTYPE > [Sub](#)
- typedef [StoragePolicy](#)< [Matrix](#) > [Storage](#)
- typedef [Vector](#)< RTYPE, [StoragePolicy](#) > [VECTOR](#)
- typedef [VECTOR::iterator](#) [iterator](#)
- typedef [VECTOR::const\\_iterator](#) [const\\_iterator](#)
- typedef [VECTOR::converter\\_type](#) [converter\\_type](#)
- typedef [VECTOR::stored\\_type](#) [stored\\_type](#)
- typedef [VECTOR::Proxy](#) [Proxy](#)
- typedef [VECTOR::const\\_Proxy](#) [const\\_Proxy](#)

## Public Member Functions

- [Matrix](#) ()
- [Matrix](#) (SEXP x)
- [Matrix](#) (const [Dimension](#) &dims)
- [Matrix](#) (const int &nrows\_, const int &ncols)
- template<typename Iterator >  
[Matrix](#) (const int &nrows\_, const int &ncols, Iterator start)
- [Matrix](#) (const int &n)

- [Matrix](#) (const [Matrix](#) &other)
- template<bool NA, typename MAT >  
[Matrix](#) (const [MatrixBase](#)< RTYPE, NA, MAT > &other)
- [Matrix](#) (const [SubMatrix](#)< RTYPE > &)
- [Matrix](#) & [operator=](#) (const [Matrix](#) &other)
- [Matrix](#) & [operator=](#) (const [SubMatrix](#)< RTYPE > &)
- [Matrix](#) (const [no\\_init\\_matrix](#) &obj)
- int [ncol](#) () const
- int [nrow](#) () const
- int [cols](#) () const
- int [rows](#) () const
- [Row](#) [row](#) (int i)
- [ConstRow](#) [row](#) (int i) const
- [Column](#) [column](#) (int i)
- [ConstColumn](#) [column](#) (int i) const
- [const\\_iterator](#) [begin](#) () const
- [const\\_iterator](#) [end](#) () const
- [const\\_iterator](#) [cbegin](#) () const
- [const\\_iterator](#) [cend](#) () const
- [iterator](#) [begin](#) ()
- [iterator](#) [end](#) ()
- template<typename U >  
void [fill\\_diag](#) (const U &u)
- [Proxy](#) [operator\[\]](#) (R\_xlen\_t i)
- [const\\_Proxy](#) [operator\[\]](#) (R\_xlen\_t i) const
- [Proxy](#) [operator\(\)](#) (const size\_t &i, const size\_t &j)
- [const\\_Proxy](#) [operator\(\)](#) (const size\_t &i, const size\_t &j) const
- [Proxy](#) [at](#) (const size\_t &i, const size\_t &j)
- [const\\_Proxy](#) [at](#) (const size\_t &i, const size\_t &j) const
- [Row](#) [operator\(\)](#) (int i, [internal::NamedPlaceholder](#))
- [ConstRow](#) [operator\(\)](#) (int i, [internal::NamedPlaceholder](#)) const
- [Column](#) [operator\(\)](#) ([internal::NamedPlaceholder](#), int i)
- [ConstColumn](#) [operator\(\)](#) ([internal::NamedPlaceholder](#), int i) const
- [Sub](#) [operator\(\)](#) (const [Range](#) &row\_range, const [Range](#) &col\_range)
- [Sub](#) [operator\(\)](#) ([internal::NamedPlaceholder](#), const [Range](#) &col\_range)
- [Sub](#) [operator\(\)](#) (const [Range](#) &row\_range, [internal::NamedPlaceholder](#))

## Static Public Member Functions

- template<typename U >  
static [Matrix](#) [diag](#) (int [size](#), const U &diag\_value)

## Private Member Functions

- R\_xlen\_t [offset](#) (const int i, const int j) const
- template<typename U >  
void [fill\\_diag\\_dispatch](#) ([traits::false\\_type](#), const U &u)
- template<typename U >  
void [fill\\_diag\\_dispatch](#) ([traits::true\\_type](#), const U &u)
- template<bool NA, typename MAT >  
void [import\\_matrix\\_expression](#) (const [MatrixBase](#)< RTYPE, NA, MAT > &other, int nr, int nc)

## Private Attributes

- int `nrows`

## Additional Inherited Members

### 6.377.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
class Rcpp::Matrix< RTYPE, StoragePolicy >
```

Definition at line 28 of file Matrix.h.

### 6.377.2 Member Typedef Documentation

#### 6.377.2.1 Column

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef MatrixColumn<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::Column
```

Definition at line 38 of file Matrix.h.

#### 6.377.2.2 const\_iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef VECTOR::const_iterator Rcpp::Matrix< RTYPE, StoragePolicy >::const_iterator
```

Definition at line 45 of file Matrix.h.

#### 6.377.2.3 const\_Proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef VECTOR::const_Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::const_Proxy
```

Definition at line 49 of file Matrix.h.

#### 6.377.2.4 ConstColumn

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef ConstMatrixColumn<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::ConstColumn
```

Definition at line 39 of file Matrix.h.

#### 6.377.2.5 ConstRow

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef ConstMatrixRow<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::ConstRow
```

Definition at line 37 of file Matrix.h.

#### 6.377.2.6 converter\_type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef VECTOR::converter_type Rcpp::Matrix< RTYPE, StoragePolicy >::converter_type
```

Definition at line 46 of file Matrix.h.

#### 6.377.2.7 iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef VECTOR::iterator Rcpp::Matrix< RTYPE, StoragePolicy >::iterator
```

Definition at line 44 of file Matrix.h.

#### 6.377.2.8 Proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef VECTOR::Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::Proxy
```

Definition at line 48 of file Matrix.h.

### 6.377.2.9 Row

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef MatrixRow<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::Row
```

Definition at line 36 of file Matrix.h.

### 6.377.2.10 Storage

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef StoragePolicy<Matrix> Rcpp::Matrix< RTYPE, StoragePolicy >::Storage
```

Definition at line 42 of file Matrix.h.

### 6.377.2.11 stored\_type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef VECTOR::stored_type Rcpp::Matrix< RTYPE, StoragePolicy >::stored_type
```

Definition at line 47 of file Matrix.h.

### 6.377.2.12 Sub

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef SubMatrix<RTYPE> Rcpp::Matrix< RTYPE, StoragePolicy >::Sub
```

Definition at line 40 of file Matrix.h.

### 6.377.2.13 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef Vector<RTYPE, StoragePolicy> Rcpp::Matrix< RTYPE, StoragePolicy >::VECTOR
```

Definition at line 43 of file Matrix.h.

## 6.377.3 Constructor & Destructor Documentation

**6.377.3.1 Matrix()** [1/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix ( ) [inline]
```

Definition at line 51 of file Matrix.h.

**6.377.3.2 Matrix()** [2/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
    SEXP x ) [inline]
```

Definition at line 53 of file Matrix.h.

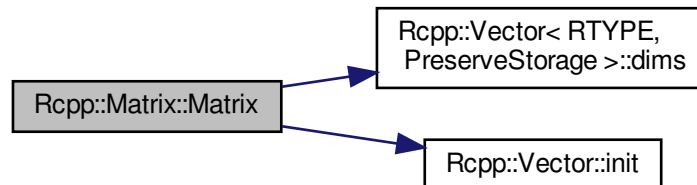
**6.377.3.3 Matrix()** [3/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
    const Dimension & dims ) [inline]
```

Definition at line 55 of file Matrix.h.

References Rcpp::Vector< RTYPE, PreserveStorage >::dims(), and Rcpp::Vector< RTYPE, StoragePolicy >::init().

Here is the call graph for this function:



**6.377.3.4 Matrix()** [4/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
    const int & nrows_,
    const int & ncols ) [inline]
```

Definition at line 59 of file Matrix.h.

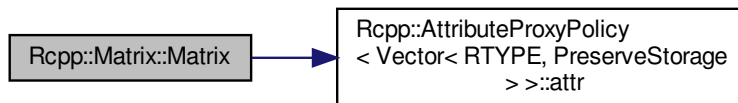
**6.377.3.5 Matrix()** [5/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename Iterator >
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
    const int & nrows_,
    const int & ncols,
    Iterator start ) [inline]
```

Definition at line 64 of file Matrix.h.

References Rcpp::AttributeProxyPolicy< Vector< RTYPE, PreserveStorage > >::attr(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

Here is the call graph for this function:

**6.377.3.6 Matrix()** [6/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
    const int & n ) [inline]
```

Definition at line 71 of file Matrix.h.

**6.377.3.7 Matrix()** [7/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
    const Matrix< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 74 of file Matrix.h.

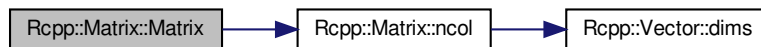
**6.377.3.8 Matrix()** [8/10]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<bool NA, typename MAT >
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
    const MatrixBase< RTYPE, NA, MAT > & other ) [inline]
```

Definition at line 77 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

Here is the call graph for this function:

**6.377.3.9 Matrix()** [9/10]

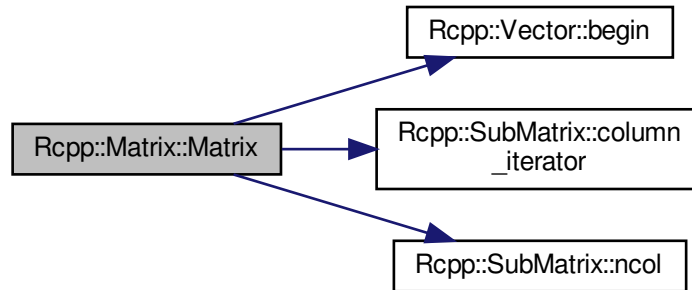
```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
    const SubMatrix< RTYPE > & sub )
```

Definition at line 59 of file SubMatrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::SubMatrix< RTYPE >::column\_iterator(), Rcpp::SubMatrix< RTYPE >::ncol(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.



Here is the call graph for this function:



### 6.377.3.10 Matrix() [10/10]

```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix (
    const no_init_matrix & obj ) [inline], [explicit]
  
```

Definition at line 92 of file Matrix.h.

## 6.377.4 Member Function Documentation

### 6.377.4.1 at() [1/2]

```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::at (
    const size_t & i,
    const size_t & j ) [inline]
  
```

Definition at line 144 of file Matrix.h.

**6.377.4.2 at()** [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::at (
    const size_t & i,
    const size_t & j ) const [inline]
```

Definition at line 147 of file Matrix.h.

**6.377.4.3 begin()** [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Matrix< RTYPE, StoragePolicy >::begin ( ) [inline]
```

Definition at line 116 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:

**6.377.4.4 begin()** [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::Matrix< RTYPE, StoragePolicy >::begin ( ) const [inline]
```

Definition at line 112 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin().

Referenced by fastLm().

Here is the call graph for this function:



### 6.377.4.5 cbegin()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>  
const_iterator Rcpp::Matrix< RTYPE, StoragePolicy >::cbegin ( ) const [inline]
```

Definition at line 114 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



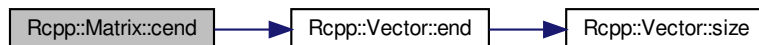
### 6.377.4.6 cend()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>  
const_iterator Rcpp::Matrix< RTYPE, StoragePolicy >::cend ( ) const [inline]
```

Definition at line 115 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



### 6.377.4.7 cols()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int Rcpp::Matrix< RTYPE, StoragePolicy >::cols ( ) const [inline]
```

Definition at line 100 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::dims().

Here is the call graph for this function:



### 6.377.4.8 column() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Column Rcpp::Matrix< RTYPE, StoragePolicy >::column (
    int i ) [inline]
```

Definition at line 109 of file Matrix.h.

### 6.377.4.9 column() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
ConstColumn Rcpp::Matrix< RTYPE, StoragePolicy >::column (
    int i ) const [inline]
```

Definition at line 110 of file Matrix.h.

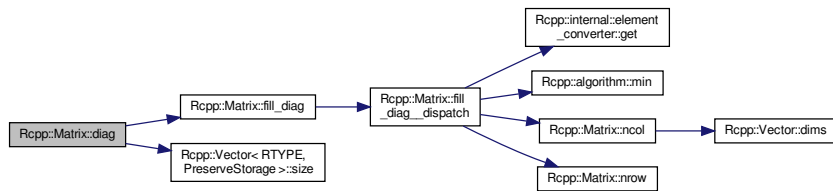
## 6.377.4.10 diag()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
static Matrix Rcpp::Matrix< RTYPE, StoragePolicy >::diag (
    int size,
    const U & diag_value ) [inline], [static]
```

Definition at line 124 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::fill\_diag(), and Rcpp::Vector< RTYPE, PreserveStorage >::size().

Here is the call graph for this function:



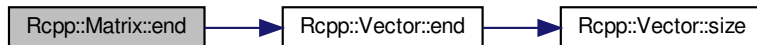
## 6.377.4.11 end() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Matrix< RTYPE, StoragePolicy >::end ( ) [inline]
```

Definition at line 117 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



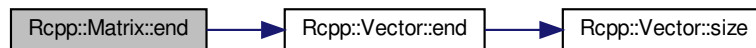
**6.377.4.12 end()** [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::Matrix< RTYPE, StoragePolicy >::end ( ) const [inline]
```

Definition at line 113 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:

**6.377.4.13 fill\_diag()**

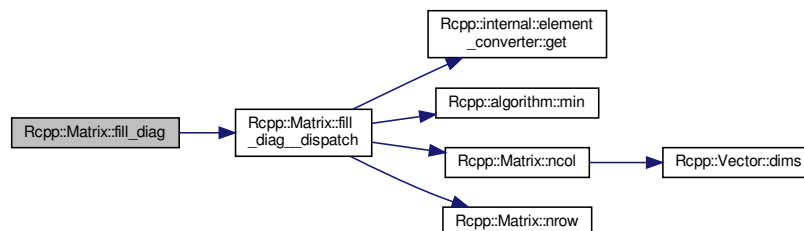
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
void Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag (
    const U & u ) [inline]
```

Definition at line 120 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::fill\_diag\_\_dispatch().

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::diag().

Here is the call graph for this function:



## 6.377.4.14 fill\_diag\_dispatch() [1/2]

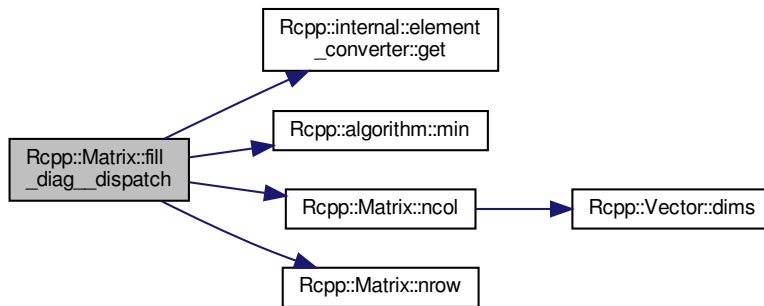
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
void Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag_dispatch (
    traits::false_type ,
    const U & u ) [inline], [private]
```

Definition at line 177 of file Matrix.h.

References Rcpp::internal::element\_converter< RTYPE >::get(), Rcpp::algorithm::min(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrow().

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::fill\_diag().

Here is the call graph for this function:



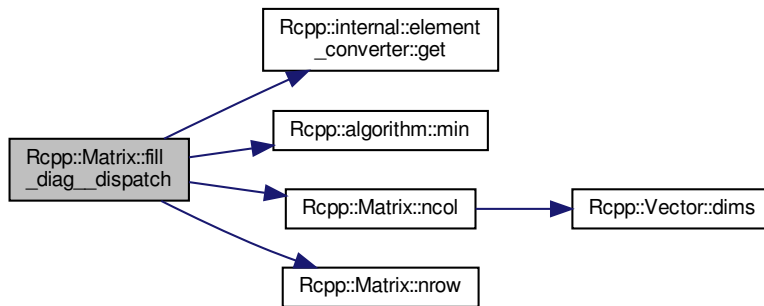
## 6.377.4.15 fill\_diag\_dispatch() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
void Rcpp::Matrix< RTYPE, StoragePolicy >::fill_diag_dispatch (
    traits::true_type ,
    const U & u ) [inline], [private]
```

Definition at line 187 of file Matrix.h.

References Rcpp::internal::element\_converter< RTYPE >::get(), Rcpp::algorithm::min(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::Matrix< RTYPE, StoragePolicy >::nrow().

Here is the call graph for this function:



#### 6.377.4.16 import\_matrix\_expression()

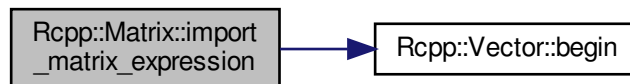
```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<bool NA, typename MAT >
void Rcpp::Matrix< RTYPE, StoragePolicy >::import_matrix_expression (
    const MatrixBase< RTYPE, NA, MAT > & other,
    int nr,
    int nc ) [inline], [private]
  
```

Definition at line 198 of file Matrix.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`.

Here is the call graph for this function:





**6.377.4.17 ncol()**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int Rcpp::Matrix< RTYPE, StoragePolicy >::ncol ( ) const [inline]
```

Definition at line 94 of file Matrix.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::dims().

Referenced by Rcpp::internal::as\_vector\_\_impl(), fastLm(), Rcpp::Matrix< RTYPE, StoragePolicy >::fill\_diag\_\_← dispatch(), Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), Rcpp::Matrix< RTYPE, StoragePolicy >::operator()(), Rcpp::MatrixRow< RTYPE >::size(), Rcpp::ConstMatrixRow< RTYPE >::size(), and Rcpp::trimws().

Here is the call graph for this function:

**6.377.4.18 nrow()**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int Rcpp::Matrix< RTYPE, StoragePolicy >::nrow ( ) const [inline]
```

Definition at line 97 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

Referenced by Rcpp::internal::as\_vector\_\_impl(), Rcpp::ConstMatrixRow< RTYPE >::ConstMatrixRow(), fastLm(), Rcpp::Matrix< RTYPE, StoragePolicy >::fill\_diag\_\_dispatch(), Rcpp::MatrixRow< RTYPE >::MatrixRow(), Rcpp::← Matrix< RTYPE, StoragePolicy >::operator()(), and Rcpp::trimws().

**6.377.4.19 offset()**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
R_xlen_t Rcpp::Matrix< RTYPE, StoragePolicy >::offset (
    const int i,
    const int j ) const [inline], [private]
```

Definition at line 174 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrows.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::operator()().

**6.377.4.20 operator>() [1/9]**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Sub Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
    const Range & row_range,
    const Range & col_range ) [inline]
```

Definition at line 163 of file Matrix.h.

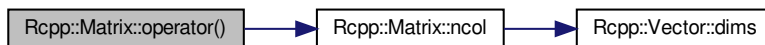
**6.377.4.21 operator>() [2/9]**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Sub Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
    const Range & row_range,
    internal::NamedPlaceholder ) [inline]
```

Definition at line 169 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::ncol().

Here is the call graph for this function:

**6.377.4.22 operator>() [3/9]**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
    const size_t & i,
    const size_t & j ) [inline]
```

Definition at line 137 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



#### 6.377.4.23 operator>() [4/9]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
    const size_t & i,
    const size_t & j ) const [inline]
```

Definition at line 140 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



#### 6.377.4.24 operator>() [5/9]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Row Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
    int i,
    internal::NamedPlaceholder ) [inline]
```

Definition at line 151 of file Matrix.h.

**6.377.4.25 operator>() [6/9]**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
ConstRow Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
    int i,
    internal::NamedPlaceholder ) const [inline]
```

Definition at line 154 of file Matrix.h.

**6.377.4.26 operator>() [7/9]**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Sub Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
    internal::NamedPlaceholder ,
    const Range & col_range ) [inline]
```

Definition at line 166 of file Matrix.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrow().

Here is the call graph for this function:

**6.377.4.27 operator>() [8/9]**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Column Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
    internal::NamedPlaceholder ,
    int i ) [inline]
```

Definition at line 157 of file Matrix.h.

**6.377.4.28 operator>() [9/9]**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
ConstColumn Rcpp::Matrix< RTYPE, StoragePolicy >::operator() (
    internal::NamedPlaceholder ,
    int i ) const [inline]
```

Definition at line 160 of file Matrix.h.

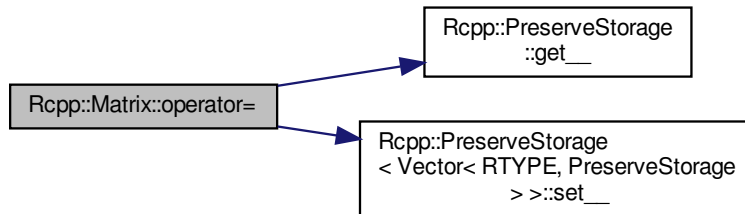
**6.377.4.29 operator=() [1/2]**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Matrix& Rcpp::Matrix< RTYPE, StoragePolicy >::operator= (
    const Matrix< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 83 of file Matrix.h.

References Rcpp::PreserveStorage< CLASS >::get\_\_(), Rcpp::Matrix< RTYPE, StoragePolicy >::nrows, and Rcpp::PreserveStorage< Vector< RTYPE, PreserveStorage > >::set\_\_().

Here is the call graph for this function:

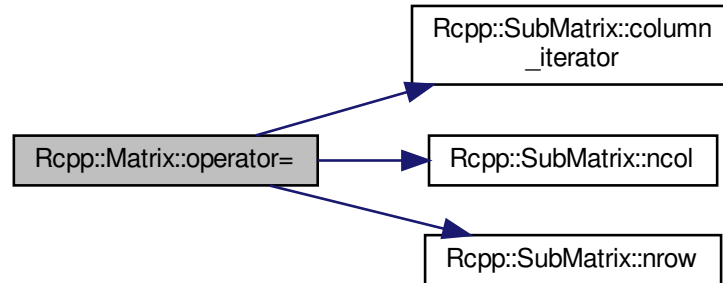
**6.377.4.30 operator=() [2/2]**

```
template<int RTYPE, template< class > class StoragePolicy>
Matrix< RTYPE, StoragePolicy > & Rcpp::Matrix< RTYPE, StoragePolicy >::operator= (
    const SubMatrix< RTYPE > & sub )
```

Definition at line 72 of file SubMatrix.h.

References `Rcpp::SubMatrix< RTYPE >::column_iterator()`, `Rcpp::SubMatrix< RTYPE >::ncol()`, and `Rcpp::SubMatrix< RTYPE >::nrow()`.

Here is the call graph for this function:



#### 6.377.4.31 `operator[]()` [1/2]

```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::operator[] (
    R_xlen_t i ) [inline]
  
```

Definition at line 130 of file `Matrix.h`.

#### 6.377.4.32 `operator[]()` [2/2]

```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_Proxy Rcpp::Matrix< RTYPE, StoragePolicy >::operator[] (
    R_xlen_t i ) const [inline]
  
```

Definition at line 133 of file `Matrix.h`.

#### 6.377.4.33 `row()` [1/2]

```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Row Rcpp::Matrix< RTYPE, StoragePolicy >::row (
    int i ) [inline]
  
```

Definition at line 107 of file `Matrix.h`.

#### 6.377.4.34 row() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
ConstRow Rcpp::Matrix< RTYPE, StoragePolicy >::row (
    int i ) const [inline]
```

Definition at line 108 of file Matrix.h.

#### 6.377.4.35 rows()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int Rcpp::Matrix< RTYPE, StoragePolicy >::rows ( ) const [inline]
```

Definition at line 103 of file Matrix.h.

References `Rcpp::Matrix< RTYPE, StoragePolicy >::nrows`.

Referenced by `Rcpp::operator<<()`.

### 6.377.5 Member Data Documentation

#### 6.377.5.1 nrows

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int Rcpp::Matrix< RTYPE, StoragePolicy >::nrows [private]
```

Definition at line 29 of file Matrix.h.

Referenced by `Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix()`, `Rcpp::Matrix< RTYPE, StoragePolicy >::nrow()`, `Rcpp::Matrix< RTYPE, StoragePolicy >::offset()`, `Rcpp::Matrix< RTYPE, StoragePolicy >::operator=()`, and `Rcpp::Matrix< RTYPE, StoragePolicy >::rows()`.

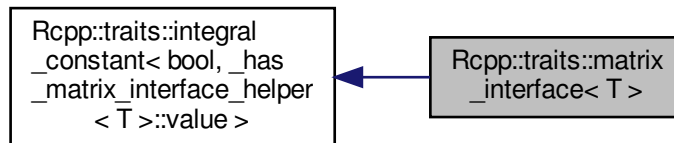
The documentation for this class was generated from the following files:

- [inst/include/Rcpp/vector/Matrix.h](#)
- [inst/include/Rcpp/vector/SubMatrix.h](#)

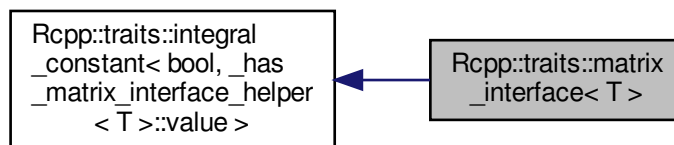
## 6.378 Rcpp::traits::matrix\_interface< T > Struct Template Reference

```
#include <matrix_interface.h>
```

Inheritance diagram for Rcpp::traits::matrix\_interface< T >:



Collaboration diagram for Rcpp::traits::matrix\_interface< T >:



### Additional Inherited Members

#### 6.378.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::matrix_interface< T >
```

Definition at line 46 of file matrix\_interface.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/matrix\_interface.h



## 6.379 Rcpp::sugar::cbind\_impl::detail::matrix\_return< T, is\_container > Struct Template Reference

```
#include <cbind.h>
```

### 6.379.1 Detailed Description

```
template<typename T, bool is_container = has_stored_type<T>::value>
struct Rcpp::sugar::cbind_impl::detail::matrix_return< T, is_container >
```

Definition at line 463 of file cbind.h.

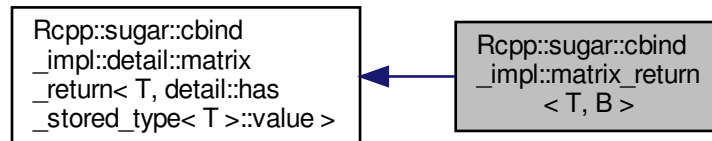
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

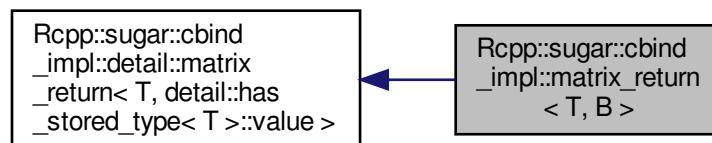
## 6.380 Rcpp::sugar::cbind\_impl::matrix\_return< T, B > Struct Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind\_impl::matrix\_return< T, B >:



Collaboration diagram for Rcpp::sugar::cbind\_impl::matrix\_return< T, B >:



### 6.380.1 Detailed Description

```
template<typename T, bool B = detail::has_stored_type<T>::value>
struct Rcpp::sugar::cbind_impl::matrix_return< T, B >
```

Definition at line 501 of file cbind.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/cbind.h](#)

## 6.381 Rcpp::sugar::cbind\_impl::detail::matrix\_return< bool, false > Struct Reference

```
#include <cbind.h>
```

### Public Types

- typedef [Rcpp::Matrix< LGLSXP > type](#)

### 6.381.1 Detailed Description

Definition at line 488 of file cbind.h.

### 6.381.2 Member Typedef Documentation

#### 6.381.2.1 type

```
typedef Rcpp::Matrix<LGLSXP> Rcpp::sugar::cbind\_impl::detail::matrix\_return< bool, false >::type
```

Definition at line 489 of file cbind.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/cbind.h](#)

## 6.382 Rcpp::sugar::cbind\_impl::detail::matrix\_return< Rcpp::Matrix< LGLSXP >, true > Struct Reference

```
#include <cbind.h>
```

### Public Types

- typedef [Rcpp::Matrix< LGLSXP >](#) [type](#)

#### 6.382.1 Detailed Description

Definition at line 478 of file cbind.h.

#### 6.382.2 Member Typedef Documentation

##### 6.382.2.1 type

```
typedef Rcpp::Matrix<LGLSXP> Rcpp::sugar::cbind\_impl::detail::matrix\_return< Rcpp::Matrix< LGLSXP >, true >::type
```

Definition at line 479 of file cbind.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/[cbind.h](#)

## 6.383 Rcpp::sugar::cbind\_impl::detail::matrix\_return< Rcpp::Vector< LGLSXP >, true > Struct Reference

```
#include <cbind.h>
```

### Public Types

- typedef [Rcpp::Matrix< LGLSXP >](#) [type](#)

#### 6.383.1 Detailed Description

Definition at line 483 of file cbind.h.

## 6.383.2 Member Typedef Documentation

### 6.383.2.1 type

```
typedef Rcpp::Matrix<LGLSXP> Rcpp::sugar::cbind_impl::detail::matrix_return< Rcpp::Vector< LGLSXP
>, true >::type
```

Definition at line 484 of file cbind.h.

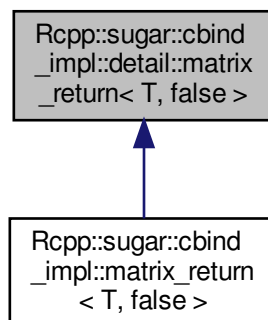
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

## 6.384 Rcpp::sugar::cbind\_impl::detail::matrix\_return< T, false > Struct Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind\_impl::detail::matrix\_return< T, false >:



### Public Types

- enum { `RTYPE` = `cbind_sexptype_traits<T>::rtype` }
- typedef `Rcpp::Matrix< RTYPE > type`

### 6.384.1 Detailed Description

```
template<typename T>
struct Rcpp::sugar::cbind_impl::detail::matrix_return< T, false >
```

Definition at line 493 of file cbind.h.

### 6.384.2 Member Typedef Documentation

#### 6.384.2.1 type

```
template<typename T >
typedef Rcpp::Matrix<RTYPE> Rcpp::sugar::cbind_impl::detail::matrix_return< T, false >::type
```

Definition at line 495 of file cbind.h.

### 6.384.3 Member Enumeration Documentation

#### 6.384.3.1 anonymous enum

```
template<typename T >
anonymous enum
```

##### Enumerator

RTYPE
-------

Definition at line 494 of file cbind.h.

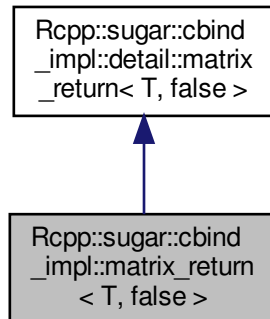
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/[cbind.h](#)

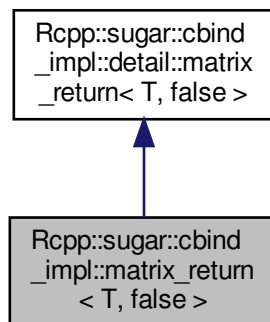
## 6.385 Rcpp::sugar::cbind\_impl::matrix\_return< T, false > Struct Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind\_impl::matrix\_return< T, false >:



Collaboration diagram for Rcpp::sugar::cbind\_impl::matrix\_return< T, false >:



## Additional Inherited Members

### 6.385.1 Detailed Description

```

template<typename T>
struct Rcpp::sugar::cbind_impl::matrix_return< T, false >

```

Definition at line 505 of file cbind.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/[cbind.h](#)

## 6.386 Rcpp::sugar::cbind\_impl::detail::matrix\_return< T, true > Struct Template Reference

```
#include <cbind.h>
```

### Public Types

- enum { [RTYPE](#) = cbind\_sexptype\_traits<stored\_type>::rtype }
- typedef [cbind\\_storage\\_type](#)< [cbind\\_sexptype\\_traits](#)< typename T::stored\_type >::rtype >::type [stored\\_type](#)
- typedef [Rcpp::Matrix](#)< [RTYPE](#) > [type](#)

### 6.386.1 Detailed Description

```
template<typename T>  
struct Rcpp::sugar::cbind_impl::detail::matrix_return< T, true >
```

Definition at line 466 of file cbind.h.

### 6.386.2 Member Typedef Documentation

#### 6.386.2.1 stored\_type

```
template<typename T >  
typedef cbind\_storage\_type< cbind\_sexptype\_traits<typename T::stored_type>::rtype >::type Rcpp::sugar::cbind\_imp  
T, true >::stored\_type
```

Definition at line 469 of file cbind.h.

#### 6.386.2.2 type

```
template<typename T >  
typedef Rcpp::Matrix<RTYPE> Rcpp::sugar::cbind\_impl::detail::matrix\_return< T, true >::type
```

Definition at line 472 of file cbind.h.

### 6.386.3 Member Enumeration Documentation

#### 6.386.3.1 anonymous enum

```
template<typename T >  
anonymous enum
```

## Enumerator

RTYPE	
-------	--

Definition at line 471 of file cbind.h.

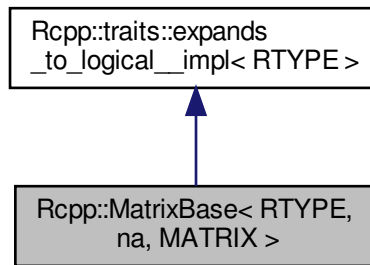
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/cbind.h](#)

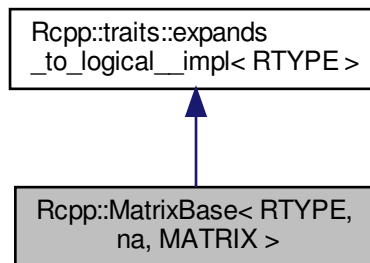
## 6.387 Rcpp::MatrixBase< RTYPE, na, MATRIX > Class Template Reference

```
#include <MatrixBase.h>
```

Inheritance diagram for Rcpp::MatrixBase< RTYPE, na, MATRIX >:



Collaboration diagram for Rcpp::MatrixBase< RTYPE, na, MATRIX >:





## Classes

- struct [can\\_have\\_na](#)
- class [iterator](#)
- struct [r\\_matrix\\_interface](#)
- struct [r\\_type](#)

## Public Types

- typedef [traits::storage\\_type](#)< RTYPE >::type [stored\\_type](#)

## Public Member Functions

- MATRIX & [get\\_ref](#) ()
- [stored\\_type operator\(\)](#) (int i, int j) const
- R\_xlen\_t [size](#) () const
- R\_xlen\_t [nrow](#) () const
- R\_xlen\_t [ncol](#) () const
- [iterator begin](#) () const
- [iterator end](#) () const

## Static Public Member Functions

- static MATRIX [eye](#) (int n)
- static MATRIX [ones](#) (int n)
- static MATRIX [zeros](#) (int n)

### 6.387.1 Detailed Description

```
template<int RTYPE, bool na, typename MATRIX>
class Rcpp::MatrixBase< RTYPE, na, MATRIX >
```

a base class for vectors, modelled after the CRTP

Definition at line 32 of file MatrixBase.h.

### 6.387.2 Member Typedef Documentation

#### 6.387.2.1 stored\_type

```
template<int RTYPE, bool na, typename MATRIX >
typedef traits::storage\_type<RTYPE>::type Rcpp::MatrixBase< RTYPE, na, MATRIX >::stored_type
```

Definition at line 37 of file MatrixBase.h.

## 6.387.3 Member Function Documentation

### 6.387.3.1 begin()

```
template<int RTYPE, bool na, typename MATRIX >  
iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::begin ( ) const [inline]
```

Definition at line 191 of file MatrixBase.h.

### 6.387.3.2 end()

```
template<int RTYPE, bool na, typename MATRIX >  
iterator Rcpp::MatrixBase< RTYPE, na, MATRIX >::end ( ) const [inline]
```

Definition at line 192 of file MatrixBase.h.

References `Rcpp::MatrixBase< RTYPE, na, MATRIX >::size()`.

Here is the call graph for this function:



### 6.387.3.3 eye()

```
template<int RTYPE, bool na, typename MATRIX >
static MATRIX Rcpp::MatrixBase< RTYPE, na, MATRIX >::eye (
    int n ) [inline], [static]
```

Definition at line 51 of file MatrixBase.h.

References Rcpp::diag().

Here is the call graph for this function:



### 6.387.3.4 get\_ref()

```
template<int RTYPE, bool na, typename MATRIX >
MATRIX& Rcpp::MatrixBase< RTYPE, na, MATRIX >::get_ref ( ) [inline]
```

Definition at line 39 of file MatrixBase.h.

Referenced by Rcpp::internal::as\_vector\_\_impl().

### 6.387.3.5 ncol()

```
template<int RTYPE, bool na, typename MATRIX >
R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol ( ) const [inline]
```

Definition at line 49 of file MatrixBase.h.

Referenced by Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA\_RM >::get(), Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA\_RM >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA\_RM >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA\_RM >::get(), and Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get().

### 6.387.3.6 nrow()

```
template<int RTYPE, bool na, typename MATRIX >
R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow ( ) const [inline]
```

Definition at line 48 of file MatrixBase.h.

Referenced by Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA\_RM >::get(), Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA\_RM >::get(), Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA\_RM >::get(), Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get(), Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA\_RM >::get(), and Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >::get().

### 6.387.3.7 ones()

```
template<int RTYPE, bool na, typename MATRIX >
static MATRIX Rcpp::MatrixBase< RTYPE, na, MATRIX >::ones (
    int n ) [inline], [static]
```

Definition at line 60 of file MatrixBase.h.

### 6.387.3.8 operator()()

```
template<int RTYPE, bool na, typename MATRIX >
stored_type Rcpp::MatrixBase< RTYPE, na, MATRIX >::operator() (
    int i,
    int j ) const [inline]
```

Definition at line 43 of file MatrixBase.h.

### 6.387.3.9 size()

```
template<int RTYPE, bool na, typename MATRIX >
R_xlen_t Rcpp::MatrixBase< RTYPE, na, MATRIX >::size ( ) const [inline]
```

Definition at line 47 of file MatrixBase.h.

Referenced by Rcpp::MatrixBase< RTYPE, na, MATRIX >::end().

### 6.387.3.10 zeros()

```
template<int RTYPE, bool na, typename MATRIX >
static MATRIX Rcpp::MatrixBase< RTYPE, na, MATRIX >::zeros (
    int n ) [inline], [static]
```

Definition at line 74 of file MatrixBase.h.

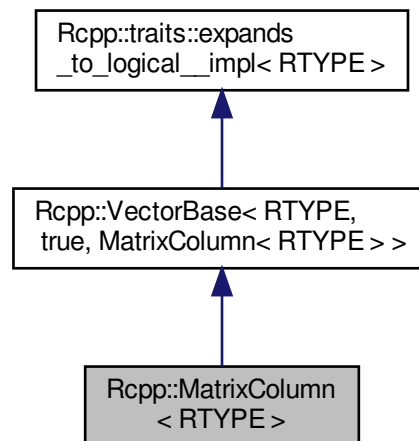
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/[MatrixBase.h](#)

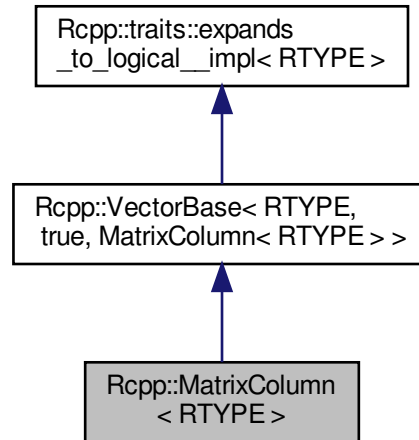
## 6.388 Rcpp::MatrixColumn< RTYPE > Class Template Reference

```
#include <MatrixColumn.h>
```

Inheritance diagram for Rcpp::MatrixColumn< RTYPE >:



Collaboration diagram for `Rcpp::MatrixColumn< RTYPE >`:



## Public Types

- typedef `Matrix< RTYPE > MATRIX`
- typedef `MATRIX::Proxy Proxy`
- typedef `MATRIX::const_Proxy const_Proxy`
- typedef `MATRIX::value_type value_type`
- typedef `MATRIX::iterator iterator`
- typedef `MATRIX::const_iterator const_iterator`

## Public Member Functions

- `MatrixColumn (MATRIX &parent, int i)`
- `MatrixColumn (const MATRIX &parent, int i)`
- `MatrixColumn (const MatrixColumn &other)`
- `template<int RT, bool NA, typename T > MatrixColumn & operator= (const Rcpp::VectorBase< RT, NA, T > &rhs)`
- `MatrixColumn & operator= (const MatrixColumn &rhs)`
- `Proxy operator[] (int i)`
- `const_Proxy operator[] (int i) const`
- `const_iterator begin () const`
- `const_iterator end () const`
- `const_iterator cbegin () const`
- `const_iterator cend () const`
- `iterator begin ()`
- `iterator end ()`
- `int size () const`

## Private Attributes

- `const int n`
- `iterator start`
- `const_iterator const_start`

### 6.388.1 Detailed Description

```
template<int RTYPE>  
class Rcpp::MatrixColumn< RTYPE >
```

Definition at line 28 of file MatrixColumn.h.

### 6.388.2 Member Typedef Documentation

#### 6.388.2.1 `const_iterator`

```
template<int RTYPE>  
typedef MATRIX::const_iterator Rcpp::MatrixColumn< RTYPE >::const_iterator
```

Definition at line 35 of file MatrixColumn.h.

#### 6.388.2.2 `const_Proxy`

```
template<int RTYPE>  
typedef MATRIX::const_Proxy Rcpp::MatrixColumn< RTYPE >::const_Proxy
```

Definition at line 32 of file MatrixColumn.h.

#### 6.388.2.3 `iterator`

```
template<int RTYPE>  
typedef MATRIX::iterator Rcpp::MatrixColumn< RTYPE >::iterator
```

Definition at line 34 of file MatrixColumn.h.

### 6.388.2.4 MATRIX

```
template<int RTYPE>
typedef MATRIX<RTYPE> Rcpp::MatrixColumn< RTYPE >::MATRIX
```

Definition at line 30 of file MatrixColumn.h.

### 6.388.2.5 Proxy

```
template<int RTYPE>
typedef MATRIX::Proxy Rcpp::MatrixColumn< RTYPE >::Proxy
```

Definition at line 31 of file MatrixColumn.h.

### 6.388.2.6 value\_type

```
template<int RTYPE>
typedef MATRIX::value_type Rcpp::MatrixColumn< RTYPE >::value_type
```

Definition at line 33 of file MatrixColumn.h.

## 6.388.3 Constructor & Destructor Documentation

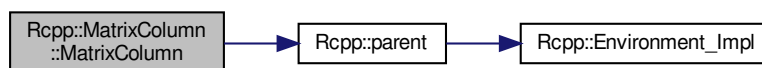
### 6.388.3.1 MatrixColumn() [1/3]

```
template<int RTYPE>
Rcpp::MatrixColumn< RTYPE >::MatrixColumn (
    MATRIX & parent,
    int i ) [inline]
```

Definition at line 37 of file MatrixColumn.h.

References Rcpp::parent().

Here is the call graph for this function:





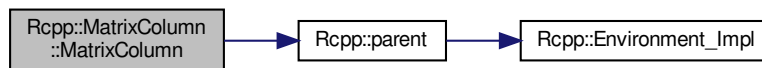
### 6.388.3.2 MatrixColumn() [2/3]

```
template<int RTYPE>
Rcpp::MatrixColumn< RTYPE >::MatrixColumn (
    const MATRIX & parent,
    int i ) [inline]
```

Definition at line 49 of file MatrixColumn.h.

References Rcpp::parent().

Here is the call graph for this function:



### 6.388.3.3 MatrixColumn() [3/3]

```
template<int RTYPE>
Rcpp::MatrixColumn< RTYPE >::MatrixColumn (
    const MatrixColumn< RTYPE > & other ) [inline]
```

Definition at line 61 of file MatrixColumn.h.

## 6.388.4 Member Function Documentation

### 6.388.4.1 begin() [1/2]

```
template<int RTYPE>
iterator Rcpp::MatrixColumn< RTYPE >::begin ( ) [inline]
```

Definition at line 103 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::start.

### 6.388.4.2 begin() [2/2]

```
template<int RTYPE>
const_iterator Rcpp::MatrixColumn< RTYPE >::begin ( ) const [inline]
```

Definition at line 87 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const\_start.

### 6.388.4.3 cbegin()

```
template<int RTYPE>
const_iterator Rcpp::MatrixColumn< RTYPE >::cbegin ( ) const [inline]
```

Definition at line 95 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const\_start.

### 6.388.4.4 cend()

```
template<int RTYPE>
const_iterator Rcpp::MatrixColumn< RTYPE >::cend ( ) const [inline]
```

Definition at line 99 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const\_start, and Rcpp::MatrixColumn< RTYPE >::n.

### 6.388.4.5 end() [1/2]

```
template<int RTYPE>
iterator Rcpp::MatrixColumn< RTYPE >::end ( ) [inline]
```

Definition at line 107 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::n, and Rcpp::MatrixColumn< RTYPE >::start.

**6.388.4.6 end()** [2/2]

```
template<int RTYPE>
const_iterator Rcpp::MatrixColumn< RTYPE >::end ( ) const [inline]
```

Definition at line 91 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const\_start, and Rcpp::MatrixColumn< RTYPE >::n.

**6.388.4.7 operator=()** [1/2]

```
template<int RTYPE>
MatrixColumn& Rcpp::MatrixColumn< RTYPE >::operator= (
    const MatrixColumn< RTYPE > & rhs ) [inline]
```

Definition at line 73 of file MatrixColumn.h.

References RCPP\_LOOP\_UNROLL, and Rcpp::MatrixColumn< RTYPE >::start.

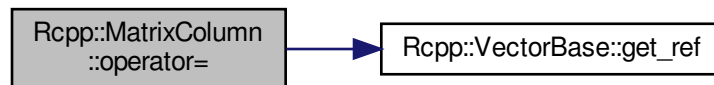
**6.388.4.8 operator=()** [2/2]

```
template<int RTYPE>
template<int RT, bool NA, typename T >
MatrixColumn& Rcpp::MatrixColumn< RTYPE >::operator= (
    const Rcpp::VectorBase< RT, NA, T > & rhs ) [inline]
```

Definition at line 67 of file MatrixColumn.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref(), RCPP\_LOOP\_UNROLL, and Rcpp::MatrixColumn< RTYPE >::start.

Here is the call graph for this function:



#### 6.388.4.9 operator[]() [1/2]

```
template<int RTYPE>
Proxy Rcpp::MatrixColumn< RTYPE >::operator[] (
    int i ) [inline]
```

Definition at line 79 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::start.

#### 6.388.4.10 operator[]() [2/2]

```
template<int RTYPE>
const_Proxy Rcpp::MatrixColumn< RTYPE >::operator[] (
    int i ) const [inline]
```

Definition at line 83 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::const\_start.

#### 6.388.4.11 size()

```
template<int RTYPE>
int Rcpp::MatrixColumn< RTYPE >::size ( ) const [inline]
```

Definition at line 111 of file MatrixColumn.h.

References Rcpp::MatrixColumn< RTYPE >::n.

### 6.388.5 Member Data Documentation

#### 6.388.5.1 const\_start

```
template<int RTYPE>
const_iterator Rcpp::MatrixColumn< RTYPE >::const_start [private]
```

Definition at line 118 of file MatrixColumn.h.

Referenced by Rcpp::MatrixColumn< RTYPE >::begin(), Rcpp::MatrixColumn< RTYPE >::cbegin(), Rcpp::MatrixColumn< RTYPE >::cend(), Rcpp::MatrixColumn< RTYPE >::end(), and Rcpp::MatrixColumn< RTYPE >::operator[]().

**6.388.5.2 n**

```
template<int RTYPE>
const int Rcpp::MatrixColumn< RTYPE >::n [private]
```

Definition at line 116 of file MatrixColumn.h.

Referenced by Rcpp::MatrixColumn< RTYPE >::end(), Rcpp::MatrixColumn< RTYPE >::end(), and Rcpp::MatrixColumn< RTYPE >::size().

**6.388.5.3 start**

```
template<int RTYPE>
iterator Rcpp::MatrixColumn< RTYPE >::start [private]
```

Definition at line 117 of file MatrixColumn.h.

Referenced by Rcpp::MatrixColumn< RTYPE >::begin(), Rcpp::MatrixColumn< RTYPE >::end(), Rcpp::MatrixColumn< RTYPE >::operator=(), and Rcpp::MatrixColumn< RTYPE >::operator[]().

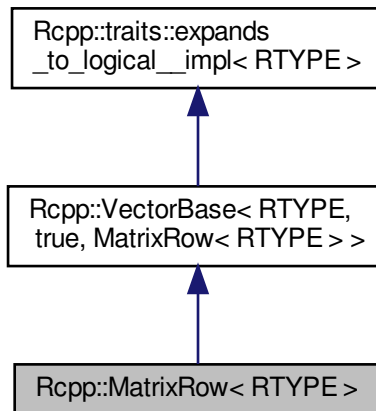
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/MatrixColumn.h

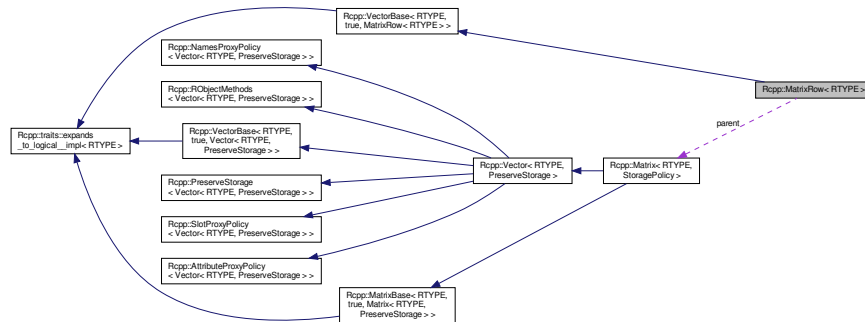
**6.389 Rcpp::MatrixRow< RTYPE > Class Template Reference**

```
#include <MatrixRow.h>
```

Inheritance diagram for Rcpp::MatrixRow< RTYPE >:



Collaboration diagram for `Rcpp::MatrixRow< RTYPE >`:



## Classes

- struct [const\\_iter\\_traits](#)
- class [iter\\_base](#)
- struct [iter\\_traits](#)

## Public Types

- typedef [Matrix< RTYPE > MATRIX](#)
- typedef [MATRIX::Proxy Proxy](#)
- typedef [MATRIX::Proxy reference](#)
- typedef [MATRIX::const\\_Proxy const\\_reference](#)
- typedef [MATRIX::value\\_type value\\_type](#)
- typedef [iter\\_base< iter\\_traits > iterator](#)
- typedef [iter\\_base< const\\_iter\\_traits > const\\_iterator](#)

## Public Member Functions

- [MatrixRow](#) ([MATRIX &object](#), int i)
- [MatrixRow](#) (const [MatrixRow](#) &other)
- template<int RT, bool NA, typename T >  
[MatrixRow](#) & [operator=](#) (const [Rcpp::VectorBase](#)< RT, NA, T > &rhs)
- [MatrixRow](#) & [operator=](#) (const [MatrixRow](#) &rhs)
- [reference operator\[\]](#) (int i)
- [reference operator\[\]](#) (int i) const
- [iterator begin](#) ()
- [iterator end](#) ()
- [const\\_iterator begin](#) () const
- [const\\_iterator end](#) () const
- [const\\_iterator cbegin](#) () const
- [const\\_iterator cend](#) () const
- int [size](#) () const

## Private Member Functions

- [int get\\_parent\\_index](#) (int i) const

## Private Attributes

- [MATRIX & parent](#)
- [MATRIX::iterator start](#)
- [int parent\\_nrow](#)
- [int row](#)

### 6.389.1 Detailed Description

```
template<int RTYPE>  
class Rcpp::MatrixRow< RTYPE >
```

Definition at line 28 of file MatrixRow.h.

### 6.389.2 Member Typedef Documentation

#### 6.389.2.1 const\_iterator

```
template<int RTYPE>  
typedef iter\_base< const\_iter\_traits > Rcpp::MatrixRow< RTYPE >::const\_iterator
```

Definition at line 131 of file MatrixRow.h.

#### 6.389.2.2 const\_reference

```
template<int RTYPE>  
typedef MATRIX::const\_Proxy Rcpp::MatrixRow< RTYPE >::const\_reference
```

Definition at line 33 of file MatrixRow.h.

### 6.389.2.3 iterator

```
template<int RTYPE>
typedef iter_base< iter_traits > Rcpp::MatrixRow< RTYPE >::iterator
```

Definition at line 130 of file MatrixRow.h.

### 6.389.2.4 MATRIX

```
template<int RTYPE>
typedef Matrix<RTYPE> Rcpp::MatrixRow< RTYPE >::MATRIX
```

Definition at line 30 of file MatrixRow.h.

### 6.389.2.5 Proxy

```
template<int RTYPE>
typedef MATRIX::Proxy Rcpp::MatrixRow< RTYPE >::Proxy
```

Definition at line 31 of file MatrixRow.h.

### 6.389.2.6 reference

```
template<int RTYPE>
typedef MATRIX::Proxy Rcpp::MatrixRow< RTYPE >::reference
```

Definition at line 32 of file MatrixRow.h.

### 6.389.2.7 value\_type

```
template<int RTYPE>
typedef MATRIX::value_type Rcpp::MatrixRow< RTYPE >::value_type
```

Definition at line 34 of file MatrixRow.h.

## 6.389.3 Constructor & Destructor Documentation



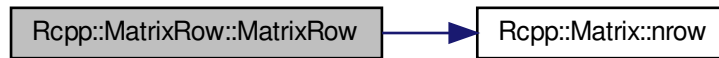
### 6.389.3.1 MatrixRow() [1/2]

```
template<int RTYPE>
Rcpp::MatrixRow< RTYPE >::MatrixRow (
    MATRIX & object,
    int i ) [inline]
```

Definition at line 133 of file MatrixRow.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::nrow(), and Rcpp::MatrixRow< RTYPE >::parent.

Here is the call graph for this function:



### 6.389.3.2 MatrixRow() [2/2]

```
template<int RTYPE>
Rcpp::MatrixRow< RTYPE >::MatrixRow (
    const MatrixRow< RTYPE > & other ) [inline]
```

Definition at line 146 of file MatrixRow.h.

## 6.389.4 Member Function Documentation

### 6.389.4.1 begin() [1/2]

```
template<int RTYPE>
iterator Rcpp::MatrixRow< RTYPE >::begin ( ) [inline]
```

Definition at line 175 of file MatrixRow.h.

**6.389.4.2 begin()** [2/2]

```
template<int RTYPE>
const_iterator Rcpp::MatrixRow< RTYPE >::begin ( ) const [inline]
```

Definition at line 183 of file MatrixRow.h.

**6.389.4.3 cbegin()**

```
template<int RTYPE>
const_iterator Rcpp::MatrixRow< RTYPE >::cbegin ( ) const [inline]
```

Definition at line 191 of file MatrixRow.h.

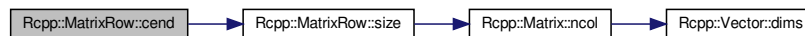
**6.389.4.4 cend()**

```
template<int RTYPE>
const_iterator Rcpp::MatrixRow< RTYPE >::cend ( ) const [inline]
```

Definition at line 195 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::size().

Here is the call graph for this function:

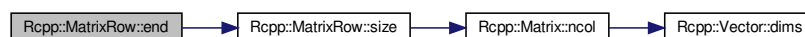
**6.389.4.5 end()** [1/2]

```
template<int RTYPE>
iterator Rcpp::MatrixRow< RTYPE >::end ( ) [inline]
```

Definition at line 179 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::size().

Here is the call graph for this function:



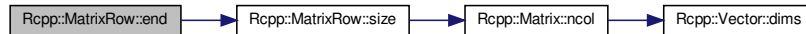
**6.389.4.6 end()** [2/2]

```
template<int RTYPE>
const_iterator Rcpp::MatrixRow< RTYPE >::end ( ) const [inline]
```

Definition at line 187 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::size().

Here is the call graph for this function:

**6.389.4.7 get\_parent\_index()**

```
template<int RTYPE>
int Rcpp::MatrixRow< RTYPE >::get_parent_index (
    int i ) const [inline], [private]
```

Definition at line 209 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::parent\_nrow, and RCPP\_DEBUG\_4.

Referenced by Rcpp::MatrixRow< RTYPE >::operator=(), and Rcpp::MatrixRow< RTYPE >::operator[]().

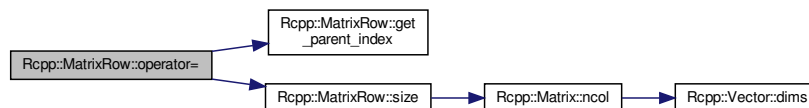
**6.389.4.8 operator=()** [1/2]

```
template<int RTYPE>
MatrixRow& Rcpp::MatrixRow< RTYPE >::operator= (
    const MatrixRow< RTYPE > & rhs ) [inline]
```

Definition at line 161 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::get\_parent\_index(), RCPP\_LOOP\_UNROLL\_LHSFUN, Rcpp::MatrixRow< RTYPE >::size(), and Rcpp::MatrixRow< RTYPE >::start.

Here is the call graph for this function:



**6.389.4.9 operator=()** [2/2]

```

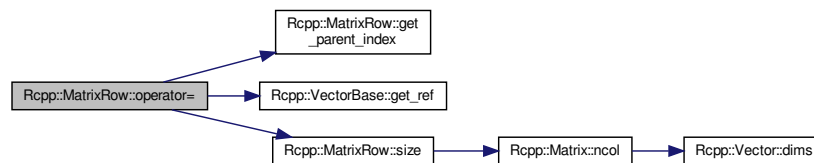
template<int RTYPE>
template<int RT, bool NA, typename T >
Rcpp::MatrixRow< RTYPE >::operator= (
    const Rcpp::VectorBase< RT, NA, T > & rhs ) [inline]

```

Definition at line 154 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::get\_parent\_index(), Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref(), RCPP\_LOOP\_UNROLL\_LHSFUN, Rcpp::MatrixRow< RTYPE >::size(), and Rcpp::MatrixRow< RTYPE >::start.

Here is the call graph for this function:

**6.389.4.10 operator[]()** [1/2]

```

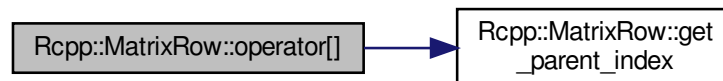
template<int RTYPE>
reference Rcpp::MatrixRow< RTYPE >::operator[] (
    int i ) [inline]

```

Definition at line 167 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::get\_parent\_index(), and Rcpp::MatrixRow< RTYPE >::start.

Here is the call graph for this function:



**6.389.4.11 operator[]()** [2/2]

```
template<int RTYPE>
reference Rcpp::MatrixRow< RTYPE >::operator[] (
    int i ) const [inline]
```

Definition at line 171 of file MatrixRow.h.

References Rcpp::MatrixRow< RTYPE >::parent, Rcpp::MatrixRow< RTYPE >::parent\_nrow, and Rcpp::MatrixRow< RTYPE >::row.

**6.389.4.12 size()**

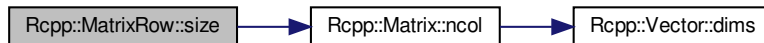
```
template<int RTYPE>
int Rcpp::MatrixRow< RTYPE >::size ( ) const [inline]
```

Definition at line 199 of file MatrixRow.h.

References Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), and Rcpp::MatrixRow< RTYPE >::parent.

Referenced by Rcpp::MatrixRow< RTYPE >::cend(), Rcpp::MatrixRow< RTYPE >::end(), and Rcpp::MatrixRow< RTYPE >::operator=().

Here is the call graph for this function:

**6.389.5 Member Data Documentation****6.389.5.1 parent**

```
template<int RTYPE>
MATRIX& Rcpp::MatrixRow< RTYPE >::parent [private]
```

Definition at line 204 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::MatrixRow(), Rcpp::MatrixRow< RTYPE >::operator[](), and Rcpp::MatrixRow< RTYPE >::size().

### 6.389.5.2 parent\_nrow

```
template<int RTYPE>
int Rcpp::MatrixRow< RTYPE >::parent_nrow [private]
```

Definition at line 206 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::get\_parent\_index(), and Rcpp::MatrixRow< RTYPE >::operator[]().

### 6.389.5.3 row

```
template<int RTYPE>
int Rcpp::MatrixRow< RTYPE >::row [private]
```

Definition at line 207 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::operator[]().

### 6.389.5.4 start

```
template<int RTYPE>
MATRIX::iterator Rcpp::MatrixRow< RTYPE >::start [private]
```

Definition at line 205 of file MatrixRow.h.

Referenced by Rcpp::MatrixRow< RTYPE >::operator=(), and Rcpp::MatrixRow< RTYPE >::operator[]().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/vector/MatrixRow.h](#)

## 6.390 Rcpp::sugar::Max< RTYPE, NA, T > Class Template Reference

```
#include <max.h>
```

### Public Types

- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) STORAGE

## Public Member Functions

- [Max](#) (const T &obj\_)
- [operator STORAGE](#) () const

## Private Attributes

- const T & [obj](#)

### 6.390.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::Max< RTYPE, NA, T >
```

Definition at line 28 of file max.h.

### 6.390.2 Member Typedef Documentation

#### 6.390.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Max< RTYPE, NA, T >::STORAGE
```

Definition at line 30 of file max.h.

### 6.390.3 Constructor & Destructor Documentation

#### 6.390.3.1 Max()

```
template<int RTYPE, bool NA, typename T >  
Rcpp::sugar::Max< RTYPE, NA, T >::Max (  
    const T & obj_ ) [inline]
```

Definition at line 32 of file max.h.

### 6.390.4 Member Function Documentation

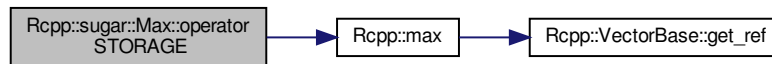
### 6.390.4.1 operator STORAGE()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Max< RTYPE, NA, T >::operator STORAGE ( ) const [inline]
```

Definition at line 34 of file max.h.

References Rcpp::max(), and Rcpp::sugar::Max< RTYPE, NA, T >::obj.

Here is the call graph for this function:



## 6.390.5 Member Data Documentation

### 6.390.5.1 obj

```
template<int RTYPE, bool NA, typename T >
const T& Rcpp::sugar::Max< RTYPE, NA, T >::obj [private]
```

Definition at line 50 of file max.h.

Referenced by Rcpp::sugar::Max< RTYPE, NA, T >::operator STORAGE(), and Rcpp::sugar::Max< RTYPE, false, T >::operator STORAGE().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/max.h](#)

## 6.391 Rcpp::sugar::Max< RTYPE, false, T > Class Template Reference

```
#include <max.h>
```

### Public Types

- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type STORAGE



## Public Member Functions

- [Max](#) (const T &obj\_)
- [operator STORAGE](#) () const

## Private Attributes

- const T & [obj](#)

### 6.391.1 Detailed Description

```
template<int RTYPE, typename T>
class Rcpp::sugar::Max< RTYPE, false, T >
```

Definition at line 55 of file max.h.

### 6.391.2 Member Typedef Documentation

#### 6.391.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Max< RTYPE, false, T >::STORAGE
```

Definition at line 57 of file max.h.

### 6.391.3 Constructor & Destructor Documentation

#### 6.391.3.1 Max()

```
template<int RTYPE, typename T >
Rcpp::sugar::Max< RTYPE, false, T >::Max (
    const T & obj_ ) [inline]
```

Definition at line 59 of file max.h.

### 6.391.4 Member Function Documentation

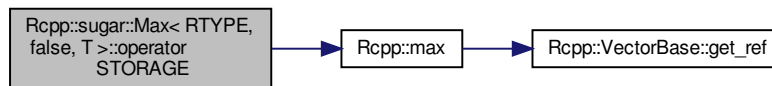
### 6.391.4.1 operator STORAGE()

```
template<int RTYPE, typename T >
Rcpp::sugar::Max< RTYPE, false, T >::operator STORAGE ( ) const [inline]
```

Definition at line 61 of file max.h.

References Rcpp::max(), and Rcpp::sugar::Max< RTYPE, NA, T >::obj.

Here is the call graph for this function:



## 6.391.5 Member Data Documentation

### 6.391.5.1 obj

```
template<int RTYPE, typename T >
const T& Rcpp::sugar::Max< RTYPE, false, T >::obj [private]
```

Definition at line 75 of file max.h.

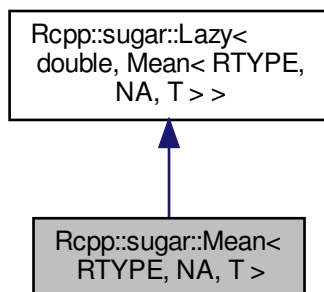
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/max.h

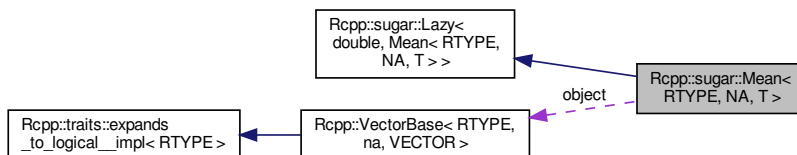
## 6.392 Rcpp::sugar::Mean< RTYPE, NA, T > Class Template Reference

```
#include <mean.h>
```

Inheritance diagram for Rcpp::sugar::Mean< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Mean< RTYPE, NA, T >:



## Public Types

- typedef [Rcpp::VectorBase< RTYPE, NA, T >](#) [VEC\\_TYPE](#)
- typedef [Rcpp::Vector< RTYPE >](#) [VECTOR](#)

## Public Member Functions

- [Mean](#) (const [VEC\\_TYPE](#) &object\_)
- double [get](#) () const

## Private Attributes

- const [VEC\\_TYPE](#) & [object](#)

### 6.392.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::Mean< RTYPE, NA, T >
```

Definition at line 29 of file mean.h.

### 6.392.2 Member Typedef Documentation

#### 6.392.2.1 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Mean< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file mean.h.

#### 6.392.2.2 VECTOR

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::Vector<RTYPE> Rcpp::sugar::Mean< RTYPE, NA, T >::VECTOR
```

Definition at line 32 of file mean.h.

### 6.392.3 Constructor & Destructor Documentation

#### 6.392.3.1 Mean()

```
template<int RTYPE, bool NA, typename T >  
Rcpp::sugar::Mean< RTYPE, NA, T >::Mean (  
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 34 of file mean.h.

### 6.392.4 Member Function Documentation

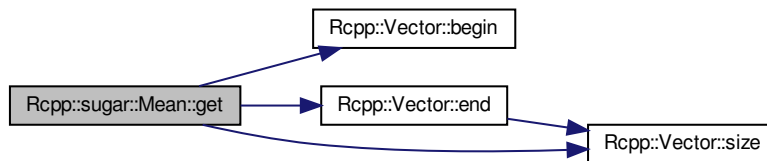
### 6.392.4.1 get()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::sugar::Mean< RTYPE, NA, T >::get ( ) const [inline]
```

Definition at line 36 of file mean.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::sugar::Mean< RTYPE, NA, T >::object, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



## 6.392.5 Member Data Documentation

### 6.392.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Mean< RTYPE, NA, T >::object [private]
```

Definition at line 51 of file mean.h.

Referenced by Rcpp::sugar::Mean< RTYPE, NA, T >::get(), Rcpp::sugar::Mean< CPLXSXP, NA, T >::get(), Rcpp::sugar::Mean< LGLSXP, NA, T >::get(), and Rcpp::sugar::Mean< INTSXP, NA, T >::get().

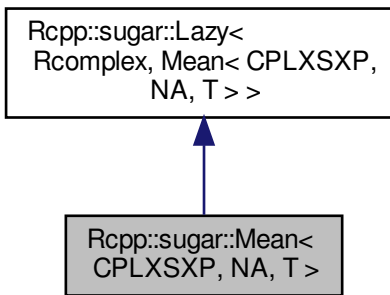
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/mean.h](#)

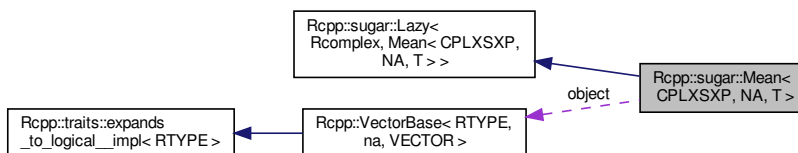
## 6.393 Rcpp::sugar::Mean< CPLXSCP, NA, T > Class Template Reference

```
#include <mean.h>
```

Inheritance diagram for Rcpp::sugar::Mean< CPLXSCP, NA, T >:



Collaboration diagram for Rcpp::sugar::Mean< CPLXSCP, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< CPLXSCP, NA, T >` `VEC_TYPE`

### Public Member Functions

- `Mean` (const `VEC_TYPE` &object\_)
- `Rcomplex` `get` () const

### Private Attributes

- const `VEC_TYPE` & `object`

## 6.393.1 Detailed Description

```
template<bool NA, typename T>
class Rcpp::sugar::Mean< CPLXSCP, NA, T >
```

Definition at line 55 of file mean.h.

## 6.393.2 Member Typedef Documentation

### 6.393.2.1 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<CPLXSCP,NA,T> Rcpp::sugar::Mean< CPLXSCP, NA, T >::VEC_TYPE
```

Definition at line 57 of file mean.h.

## 6.393.3 Constructor & Destructor Documentation

### 6.393.3.1 Mean()

```
template<bool NA, typename T >
Rcpp::sugar::Mean< CPLXSCP, NA, T >::Mean (
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 59 of file mean.h.

## 6.393.4 Member Function Documentation

### 6.393.4.1 get()

```
template<bool NA, typename T >
Rcomplex Rcpp::sugar::Mean< CPLXSXP, NA, T >::get ( ) const [inline]
```

Definition at line 61 of file mean.h.

References `Rcpp::sugar::Mean< RTYPE, NA, T >::object`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



## 6.393.5 Member Data Documentation

### 6.393.5.1 object

```
template<bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Mean< CPLXSXP, NA, T >::object [private]
```

Definition at line 88 of file mean.h.

The documentation for this class was generated from the following file:

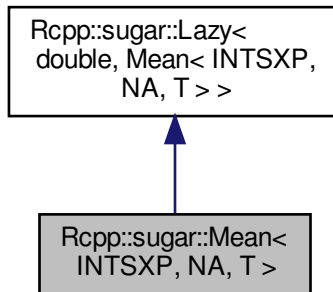
- [inst/include/Rcpp/sugar/functions/mean.h](#)



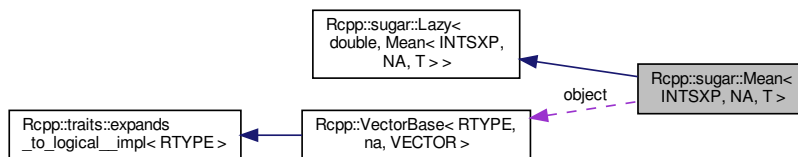
## 6.394 Rcpp::sugar::Mean< INTSXP, NA, T > Class Template Reference

```
#include <mean.h>
```

Inheritance diagram for Rcpp::sugar::Mean< INTSXP, NA, T >:



Collaboration diagram for Rcpp::sugar::Mean< INTSXP, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< INTSXP, NA, T >` `VEC_TYPE`

### Public Member Functions

- `Mean` (const `VEC_TYPE` &object\_)
- double `get` () const

### Private Attributes

- const `VEC_TYPE` & `object`

### 6.394.1 Detailed Description

```
template<bool NA, typename T>  
class Rcpp::sugar::Mean< INTSXP, NA, T >
```

Definition at line 114 of file mean.h.

### 6.394.2 Member Typedef Documentation

#### 6.394.2.1 VEC\_TYPE

```
template<bool NA, typename T >  
typedef Rcpp::VectorBase<INTSXP,NA,T> Rcpp::sugar::Mean< INTSXP, NA, T >::VEC_TYPE
```

Definition at line 116 of file mean.h.

### 6.394.3 Constructor & Destructor Documentation

#### 6.394.3.1 Mean()

```
template<bool NA, typename T >  
Rcpp::sugar::Mean< INTSXP, NA, T >::Mean (  
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 118 of file mean.h.

### 6.394.4 Member Function Documentation

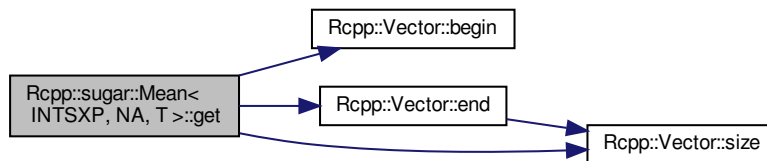
### 6.394.4.1 get()

```
template<bool NA, typename T >  
double Rcpp::sugar::Mean< INTSXP, NA, T >::get ( ) const [inline]
```

Definition at line 120 of file mean.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::sugar::Mean< RTYPE, NA, T >::object`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



## 6.394.5 Member Data Documentation

### 6.394.5.1 object

```
template<bool NA, typename T >  
const VEC_TYPE& Rcpp::sugar::Mean< INTSXP, NA, T >::object [private]
```

Definition at line 134 of file mean.h.

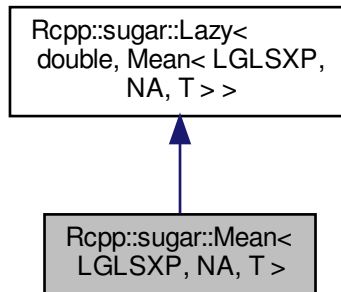
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/mean.h`

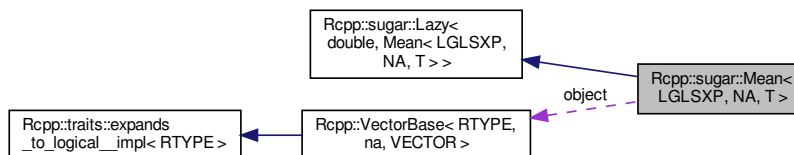
## 6.395 Rcpp::sugar::Mean< LGLSXP, NA, T > Class Template Reference

```
#include <mean.h>
```

Inheritance diagram for Rcpp::sugar::Mean< LGLSXP, NA, T >:



Collaboration diagram for Rcpp::sugar::Mean< LGLSXP, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< LGLSXP, NA, T >` `VEC_TYPE`

### Public Member Functions

- `Mean` (const `VEC_TYPE` &object\_)
- double `get` () const

### Private Attributes

- const `VEC_TYPE` & `object`

## 6.395.1 Detailed Description

```
template<bool NA, typename T>
class Rcpp::sugar::Mean< LGLSXP, NA, T >
```

Definition at line 92 of file mean.h.

## 6.395.2 Member Typedef Documentation

### 6.395.2.1 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<LGLSXP,NA,T> Rcpp::sugar::Mean< LGLSXP, NA, T >::VEC_TYPE
```

Definition at line 94 of file mean.h.

## 6.395.3 Constructor & Destructor Documentation

### 6.395.3.1 Mean()

```
template<bool NA, typename T >
Rcpp::sugar::Mean< LGLSXP, NA, T >::Mean (
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 96 of file mean.h.

## 6.395.4 Member Function Documentation

### 6.395.4.1 get()

```
template<bool NA, typename T >
double Rcpp::sugar::Mean< LGLSXP, NA, T >::get ( ) const [inline]
```

Definition at line 98 of file mean.h.

References `Rcpp::sugar::Mean< RTYPE, NA, T >::object`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



## 6.395.5 Member Data Documentation

### 6.395.5.1 object

```
template<bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Mean< LGLSXP, NA, T >::object [private]
```

Definition at line 110 of file mean.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/mean.h](#)

## 6.396 Rcpp::sugar::Median< RTYPE, NA, T, NA\_RM > Class Template Reference

```
#include <median.h>
```

## Public Types

- enum { [RESULT\\_RTYPE](#) = median\_detail::result<RTYPE>::rtype }
- typedef [median\\_detail::result](#)< RTYPE >::type [result\\_type](#)
- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type [stored\\_type](#)
- typedef T [VECTOR](#)

## Public Member Functions

- [Median](#) (const [VECTOR](#) &xx)
- [operator result\\_type](#) ()

## Private Attributes

- [VECTOR](#) x

### 6.396.1 Detailed Description

```
template<int RTYPE, bool NA, typename T, bool NA_RM = false>
class Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >
```

Definition at line 87 of file median.h.

### 6.396.2 Member Typedef Documentation

#### 6.396.2.1 result\_type

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef median\_detail::result<RTYPE>::type Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::result\_type
```

Definition at line 89 of file median.h.

#### 6.396.2.2 stored\_type

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >↔
::stored\_type
```

Definition at line 90 of file median.h.

### 6.396.2.3 VECTOR

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef T Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::VECTOR
```

Definition at line 92 of file median.h.

## 6.396.3 Member Enumeration Documentation

### 6.396.3.1 anonymous enum

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
anonymous enum
```

#### Enumerator

RESULT_RTYPE
--------------

Definition at line 91 of file median.h.

## 6.396.4 Constructor & Destructor Documentation

### 6.396.4.1 Median()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::Median (
    const VECTOR & xx ) [inline]
```

Definition at line 98 of file median.h.

## 6.396.5 Member Function Documentation



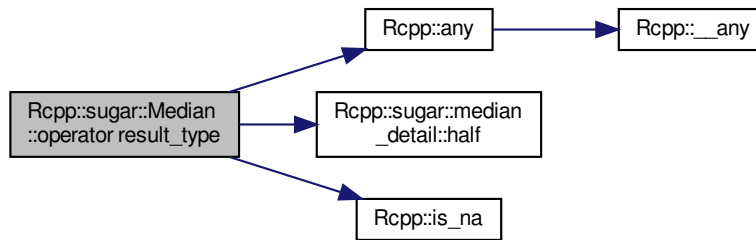
### 6.396.5.1 operator result\_type()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::operator result_type ( ) [inline]
```

Definition at line 101 of file median.h.

References [Rcpp::any\(\)](#), [Rcpp::sugar::median\\_detail::half\(\)](#), [Rcpp::is\\_na\(\)](#), and [Rcpp::sugar::Median< RTYPE, NA, T, NA\\_RM >::x](#).

Here is the call graph for this function:



## 6.396.6 Member Data Documentation

### 6.396.6.1 x

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
VECTOR Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::x [private]
```

Definition at line 95 of file median.h.

Referenced by [Rcpp::sugar::Median< RTYPE, NA, T, NA\\_RM >::operator result\\_type\(\)](#), [Rcpp::sugar::Median< RTYPE, NA, T, true >::operator result\\_type\(\)](#), [Rcpp::sugar::Median< RTYPE, false, T, NA\\_RM >::operator result\\_type\(\)](#), [Rcpp::sugar::Median< STRSXP, NA, T, NA\\_RM >::operator result\\_type\(\)](#), [Rcpp::sugar::Median< STRSXP, NA, T, true >::operator result\\_type\(\)](#), and [Rcpp::sugar::Median< STRSXP, false, T, true >::operator result\\_type\(\)](#).

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/median.h](#)

## 6.397 Rcpp::sugar::Median< RTYPE, false, T, NA\_RM > Class Template Reference

```
#include <median.h>
```

### Public Types

- enum { [RESULT\\_RTYPE](#) = median\_detail::result<RTYPE>::rtype }
- typedef [median\\_detail::result< RTYPE >::type](#) [result\\_type](#)
- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) [stored\\_type](#)
- typedef T [VECTOR](#)

### Public Member Functions

- [Median](#) (const [VECTOR](#) &xx)
- [operator result\\_type](#) ()

### Private Attributes

- [VECTOR](#) x

### 6.397.1 Detailed Description

```
template<int RTYPE, typename T, bool NA_RM>  
class Rcpp::sugar::Median< RTYPE, false, T, NA_RM >
```

Definition at line 159 of file median.h.

### 6.397.2 Member Typedef Documentation

#### 6.397.2.1 result\_type

```
template<int RTYPE, typename T , bool NA_RM>  
typedef median\_detail::result<RTYPE>::type Rcpp::sugar::Median< RTYPE, false, T, NA\_RM >::result\_type
```

Definition at line 161 of file median.h.

### 6.397.2.2 stored\_type

```
template<int RTYPE, typename T , bool NA_RM>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Median< RTYPE, false, T, NA_RM >↔
::stored_type
```

Definition at line 162 of file median.h.

### 6.397.2.3 VECTOR

```
template<int RTYPE, typename T , bool NA_RM>
typedef T Rcpp::sugar::Median< RTYPE, false, T, NA_RM >::VECTOR
```

Definition at line 164 of file median.h.

## 6.397.3 Member Enumeration Documentation

### 6.397.3.1 anonymous enum

```
template<int RTYPE, typename T , bool NA_RM>
anonymous enum
```

#### Enumerator

RESULT_RTYPE
--------------

Definition at line 163 of file median.h.

## 6.397.4 Constructor & Destructor Documentation

### 6.397.4.1 Median()

```
template<int RTYPE, typename T , bool NA_RM>
Rcpp::sugar::Median< RTYPE, false, T, NA_RM >::Median (
    const VECTOR & xx ) [inline]
```

Definition at line 170 of file median.h.

## 6.397.5 Member Function Documentation

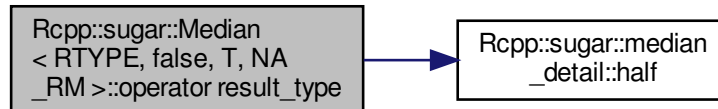
### 6.397.5.1 operator result\_type()

```
template<int RTYPE, typename T , bool NA_RM>
Rcpp::sugar::Median< RTYPE, false, T, NA_RM >::operator result_type ( ) [inline]
```

Definition at line 173 of file median.h.

References Rcpp::sugar::median\_detail::half(), and Rcpp::sugar::Median< RTYPE, NA, T, NA\_RM >::x.

Here is the call graph for this function:



## 6.397.6 Member Data Documentation

### 6.397.6.1 x

```
template<int RTYPE, typename T , bool NA_RM>
VECTOR Rcpp::sugar::Median< RTYPE, false, T, NA_RM >::x [private]
```

Definition at line 167 of file median.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[median.h](#)

## 6.398 Rcpp::sugar::Median< RTYPE, NA, T, true > Class Template Reference

```
#include <median.h>
```

## Public Types

- enum { [RESULT\\_RTYPE](#) = median\_detail::result<RTYPE>::rtype }
- typedef [median\\_detail::result](#)< RTYPE >::type [result\\_type](#)
- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type [stored\\_type](#)
- typedef T [VECTOR](#)

## Public Member Functions

- [Median](#) (const [VECTOR](#) &xx)
- [operator result\\_type](#) ()

## Private Attributes

- [VECTOR](#) x

### 6.398.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Median< RTYPE, NA, T, true >
```

Definition at line 125 of file median.h.

### 6.398.2 Member Typedef Documentation

#### 6.398.2.1 result\_type

```
template<int RTYPE, bool NA, typename T >
typedef median\_detail::result<RTYPE>::type Rcpp::sugar::Median< RTYPE, NA, T, true >::result\_type
```

Definition at line 127 of file median.h.

#### 6.398.2.2 stored\_type

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Median< RTYPE, NA, T, true >↔
::stored\_type
```

Definition at line 128 of file median.h.

### 6.398.2.3 VECTOR

```
template<int RTYPE, bool NA, typename T >  
typedef T Rcpp::sugar::Median< RTYPE, NA, T, true >::VECTOR
```

Definition at line 130 of file median.h.

## 6.398.3 Member Enumeration Documentation

### 6.398.3.1 anonymous enum

```
template<int RTYPE, bool NA, typename T >  
anonymous enum
```

#### Enumerator

RESULT_RTYPE
--------------

Definition at line 129 of file median.h.

## 6.398.4 Constructor & Destructor Documentation

### 6.398.4.1 Median()

```
template<int RTYPE, bool NA, typename T >  
Rcpp::sugar::Median< RTYPE, NA, T, true >::Median (  
    const VECTOR & xx ) [inline]
```

Definition at line 136 of file median.h.

## 6.398.5 Member Function Documentation

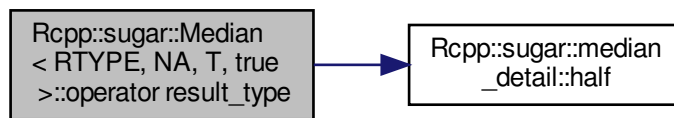
### 6.398.5.1 operator result\_type()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Median< RTYPE, NA, T, true >::operator result_type ( ) [inline]
```

Definition at line 139 of file median.h.

References Rcpp::sugar::median\_detail::half(), and Rcpp::sugar::Median< RTYPE, NA, T, NA\_RM >::x.

Here is the call graph for this function:



## 6.398.6 Member Data Documentation

### 6.398.6.1 x

```
template<int RTYPE, bool NA, typename T >
VECTOR Rcpp::sugar::Median< RTYPE, NA, T, true >::x [private]
```

Definition at line 133 of file median.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[median.h](#)

## 6.399 Rcpp::sugar::Median< STRSXP, false, T, true > Class Template Reference

```
#include <median.h>
```

## Public Types

- typedef `median_detail::result< STRSXP >::type` `result_type`
- typedef `Rcpp::traits::storage_type< STRSXP >::type` `stored_type`
- typedef T `VECTOR`

## Public Member Functions

- `Median` (const `VECTOR` &xx)
- `operator result_type` ()

## Private Attributes

- `VECTOR x`

### 6.399.1 Detailed Description

```
template<typename T>  
class Rcpp::sugar::Median< STRSXP, false, T, true >
```

Definition at line 254 of file median.h.

### 6.399.2 Member Typedef Documentation

#### 6.399.2.1 result\_type

```
template<typename T >  
typedef median_detail::result<STRSXP>::type Rcpp::sugar::Median< STRSXP, false, T, true >  
::result_type
```

Definition at line 256 of file median.h.

#### 6.399.2.2 stored\_type

```
template<typename T >  
typedef Rcpp::traits::storage_type<STRSXP>::type Rcpp::sugar::Median< STRSXP, false, T, true >  
::stored_type
```

Definition at line 257 of file median.h.



### 6.399.2.3 VECTOR

```
template<typename T >
typedef T Rcpp::sugar::Median< STRSXP, false, T, true >::VECTOR
```

Definition at line 258 of file median.h.

## 6.399.3 Constructor & Destructor Documentation

### 6.399.3.1 Median()

```
template<typename T >
Rcpp::sugar::Median< STRSXP, false, T, true >::Median (
    const VECTOR & xx ) [inline]
```

Definition at line 264 of file median.h.

## 6.399.4 Member Function Documentation

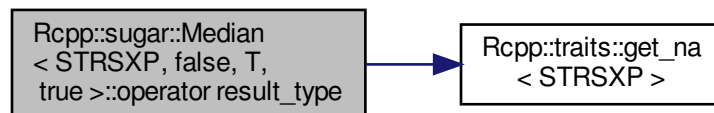
### 6.399.4.1 operator result\_type()

```
template<typename T >
Rcpp::sugar::Median< STRSXP, false, T, true >::operator result_type ( ) [inline]
```

Definition at line 267 of file median.h.

References `Rcpp::traits::get_na< STRSXP >()`, and `Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::x`.

Here is the call graph for this function:



## 6.399.5 Member Data Documentation

### 6.399.5.1 x

```
template<typename T >  
VECTOR Rcpp::sugar::Median< STRSXP, false, T, true >::x [private]
```

Definition at line 261 of file median.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/median.h](#)

## 6.400 Rcpp::sugar::Median< STRSXP, NA, T, NA\_RM > Class Template Reference

```
#include <median.h>
```

### Public Types

- typedef [median\\_detail::result< STRSXP >::type](#) [result\\_type](#)
- typedef [Rcpp::traits::storage\\_type< STRSXP >::type](#) [stored\\_type](#)
- typedef T [VECTOR](#)

### Public Member Functions

- [Median](#) (const [VECTOR](#) &xx)
- [operator result\\_type](#) ()

### Private Attributes

- [VECTOR](#) x

### 6.400.1 Detailed Description

```
template<bool NA, typename T, bool NA_RM>  
class Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >
```

Definition at line 196 of file median.h.

## 6.400.2 Member Typedef Documentation

### 6.400.2.1 result\_type

```
template<bool NA, typename T , bool NA_RM>
typedef median_detail::result<STRSXP>::type Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::result_type
```

Definition at line 198 of file median.h.

### 6.400.2.2 stored\_type

```
template<bool NA, typename T , bool NA_RM>
typedef Rcpp::traits::storage_type<STRSXP>::type Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >↔
::stored_type
```

Definition at line 199 of file median.h.

### 6.400.2.3 VECTOR

```
template<bool NA, typename T , bool NA_RM>
typedef T Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::VECTOR
```

Definition at line 200 of file median.h.

## 6.400.3 Constructor & Destructor Documentation

### 6.400.3.1 Median()

```
template<bool NA, typename T , bool NA_RM>
Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::Median (
    const VECTOR & xx ) [inline]
```

Definition at line 206 of file median.h.

## 6.400.4 Member Function Documentation

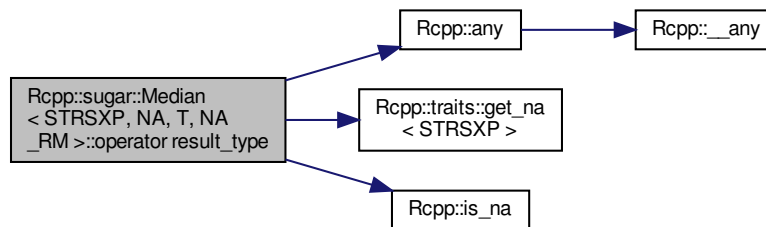
### 6.400.4.1 operator result\_type()

```
template<bool NA, typename T , bool NA_RM>
Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::operator result_type ( ) [inline]
```

Definition at line 209 of file median.h.

References `Rcpp::any()`, `Rcpp::traits::get_na< STRSXP >()`, `Rcpp::is_na()`, and `Rcpp::sugar::Median< RTYPE, NA, T, NA_RM >::x`.

Here is the call graph for this function:



## 6.400.5 Member Data Documentation

### 6.400.5.1 x

```
template<bool NA, typename T , bool NA_RM>
VECTOR Rcpp::sugar::Median< STRSXP, NA, T, NA_RM >::x [private]
```

Definition at line 203 of file median.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/median.h](#)

## 6.401 Rcpp::sugar::Median< STRSXP, NA, T, true > Class Template Reference

```
#include <median.h>
```

### Public Types

- typedef [median\\_detail::result< STRSXP >::type](#) [result\\_type](#)
- typedef [Rcpp::traits::storage\\_type< STRSXP >::type](#) [stored\\_type](#)
- typedef T [VECTOR](#)

### Public Member Functions

- [Median](#) (const [VECTOR](#) &xx)
- [operator result\\_type](#) ()

### Private Attributes

- [VECTOR](#) x

### 6.401.1 Detailed Description

```
template<bool NA, typename T>  
class Rcpp::sugar::Median< STRSXP, NA, T, true >
```

Definition at line 227 of file median.h.

### 6.401.2 Member Typedef Documentation

#### 6.401.2.1 result\_type

```
template<bool NA, typename T >  
typedef median\_detail::result<STRSXP>::type Rcpp::sugar::Median< STRSXP, NA, T, true >::result\_type
```

Definition at line 229 of file median.h.

### 6.401.2.2 stored\_type

```
template<bool NA, typename T >
typedef Rcpp::traits::storage_type<STRSXP>::type Rcpp::sugar::Median< STRSXP, NA, T, true >↔
::stored_type
```

Definition at line 230 of file median.h.

### 6.401.2.3 VECTOR

```
template<bool NA, typename T >
typedef T Rcpp::sugar::Median< STRSXP, NA, T, true >::VECTOR
```

Definition at line 231 of file median.h.

## 6.401.3 Constructor & Destructor Documentation

### 6.401.3.1 Median()

```
template<bool NA, typename T >
Rcpp::sugar::Median< STRSXP, NA, T, true >::Median (
    const VECTOR & xx ) [inline]
```

Definition at line 237 of file median.h.

## 6.401.4 Member Function Documentation

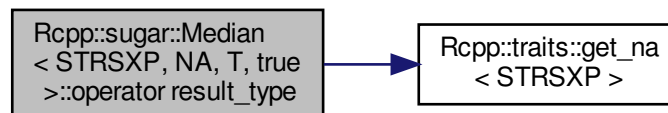
### 6.401.4.1 operator result\_type()

```
template<bool NA, typename T >
Rcpp::sugar::Median< STRSXP, NA, T, true >::operator result_type ( ) [inline]
```

Definition at line 240 of file median.h.

References Rcpp::traits::get\_na< STRSXP >(), and Rcpp::sugar::Median< RTYPE, NA, T, NA\_RM >::x.

Here is the call graph for this function:



## 6.401.5 Member Data Documentation

### 6.401.5.1 x

```
template<bool NA, typename T >  
VECTOR Rcpp::sugar::Median< STRSXP, NA, T, true >::x [private]
```

Definition at line 234 of file median.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/median.h](#)

## 6.402 Rcpp::sugar::Min< RTYPE, NA, T > Class Template Reference

```
#include <min.h>
```

### Public Types

- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) STORAGE

### Public Member Functions

- [Min](#) (const T &obj\_)
- [operator STORAGE](#) () const

### Public Attributes

- const T & [obj](#)

### 6.402.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::Min< RTYPE, NA, T >
```

Definition at line 28 of file min.h.

## 6.402.2 Member Typedef Documentation

### 6.402.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Min< RTYPE, NA, T >::STORAGE
```

Definition at line 30 of file min.h.

## 6.402.3 Constructor & Destructor Documentation

### 6.402.3.1 Min()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Min< RTYPE, NA, T >::Min (
    const T & obj_ ) [inline]
```

Definition at line 32 of file min.h.

## 6.402.4 Member Function Documentation

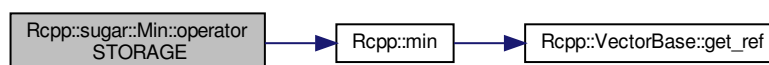
### 6.402.4.1 operator STORAGE()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Min< RTYPE, NA, T >::operator STORAGE ( ) const [inline]
```

Definition at line 34 of file min.h.

References Rcpp::min(), and Rcpp::sugar::Min< RTYPE, NA, T >::obj.

Here is the call graph for this function:





## 6.402.5 Member Data Documentation

### 6.402.5.1 obj

```
template<int RTYPE, bool NA, typename T >  
const T& Rcpp::sugar::Min< RTYPE, NA, T >::obj
```

Definition at line 50 of file min.h.

Referenced by Rcpp::sugar::Min< RTYPE, NA, T >::operator STORAGE(), and Rcpp::sugar::Min< RTYPE, false, T >::operator STORAGE().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/min.h

## 6.403 Rcpp::sugar::Min< RTYPE, false, T > Class Template Reference

```
#include <min.h>
```

### Public Types

- typedef Rcpp::traits::storage\_type< RTYPE >::type STORAGE

### Public Member Functions

- [Min](#) (const T &obj\_)
- [operator STORAGE](#) () const

### Public Attributes

- const T & [obj](#)

### 6.403.1 Detailed Description

```
template<int RTYPE, typename T>  
class Rcpp::sugar::Min< RTYPE, false, T >
```

Definition at line 55 of file min.h.

## 6.403.2 Member Typedef Documentation

### 6.403.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Min< RTYPE, false, T >::STORAGE
```

Definition at line 57 of file min.h.

## 6.403.3 Constructor & Destructor Documentation

### 6.403.3.1 Min()

```
template<int RTYPE, typename T >
Rcpp::sugar::Min< RTYPE, false, T >::Min (
    const T & obj_ ) [inline]
```

Definition at line 59 of file min.h.

## 6.403.4 Member Function Documentation

### 6.403.4.1 operator STORAGE()

```
template<int RTYPE, typename T >
Rcpp::sugar::Min< RTYPE, false, T >::operator STORAGE ( ) const [inline]
```

Definition at line 61 of file min.h.

References Rcpp::min(), and Rcpp::sugar::Min< RTYPE, NA, T >::obj.

Here is the call graph for this function:



## 6.403.5 Member Data Documentation

### 6.403.5.1 obj

```
template<int RTYPE, typename T >
const T& Rcpp::sugar::Min< RTYPE, false, T >::obj
```

Definition at line 74 of file min.h.

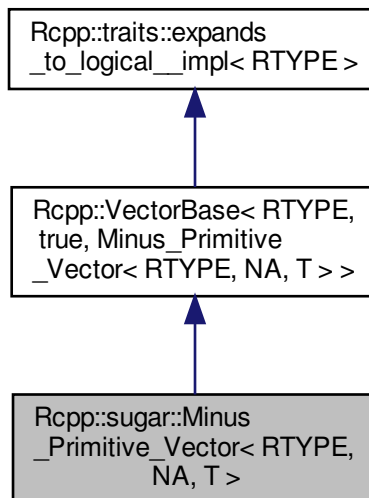
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/min.h

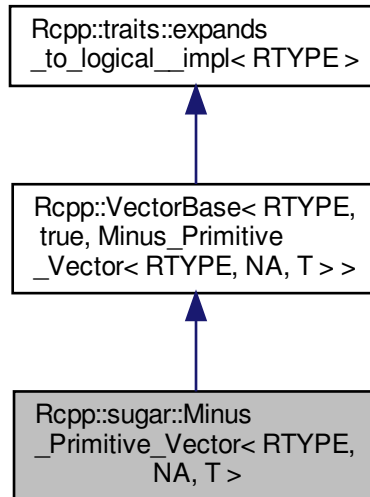
## 6.404 Rcpp::sugar::Minus\_Primitive\_Vector< RTYPE, NA, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Primitive\_Vector< RTYPE, NA, T >:



Collaboration diagram for `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >`:



## Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::traits::Extractor< RTYPE, NA, T >::type` `VEC_EXT`

## Public Member Functions

- `Minus_Primitive_Vector` (`STORAGE lhs_`, `const VEC_TYPE &rhs_`)
- `STORAGE operator[]` (`R_xlen_t i`) `const`
- `R_xlen_t size` () `const`

## Private Attributes

- `STORAGE lhs`
- `const VEC_EXT & rhs`
- `bool lhs_na`

### 6.404.1 Detailed Description

```

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >

```

Definition at line 324 of file `minus.h`.

## 6.404.2 Member Typedef Documentation

### 6.404.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >↔
::STORAGE
```

Definition at line 328 of file minus.h.

### 6.404.2.2 VEC\_EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor<RTYPE,NA,T>::type Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA,
T >::VEC_EXT
```

Definition at line 329 of file minus.h.

### 6.404.2.3 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 327 of file minus.h.

## 6.404.3 Constructor & Destructor Documentation

### 6.404.3.1 Minus\_Primitive\_Vector()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::Minus_Primitive_Vector (
    STORAGE lhs_,
    const VEC_TYPE & rhs_ ) [inline]
```

Definition at line 331 of file minus.h.

## 6.404.4 Member Function Documentation

### 6.404.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 334 of file minus.h.

References `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs`, `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs_na`, and `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs`.

### 6.404.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 338 of file minus.h.

References `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs`.

## 6.404.5 Member Data Documentation

### 6.404.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs [private]
```

Definition at line 341 of file minus.h.

Referenced by `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::operator[]()`, `Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::operator[]()`, `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::operator[]()`, and `Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::operator[]()`.

### 6.404.5.2 lhs\_na

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs_na [private]
```

Definition at line 343 of file minus.h.

Referenced by Rcpp::sugar::Minus\_Primitive\_Vector< RTYPE, NA, T >::operator[](), and Rcpp::sugar::Minus\_↵ Primitive\_Vector< RTYPE, false, T >::operator[]().

### 6.404.5.3 rhs

```
template<int RTYPE, bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs [private]
```

Definition at line 342 of file minus.h.

Referenced by Rcpp::sugar::Minus\_Primitive\_Vector< RTYPE, NA, T >::operator[](), Rcpp::sugar::Minus\_Primitive\_↵ \_Vector< REALSXP, NA, T >::operator[](), Rcpp::sugar::Minus\_Primitive\_Vector< RTYPE, false, T >::operator[](), Rcpp::sugar::Minus\_Primitive\_Vector< REALSXP, false, T >::operator[](), Rcpp::sugar::Minus\_Primitive\_Vector< RTYPE, NA, T >::size(), Rcpp::sugar::Minus\_Primitive\_Vector< REALSXP, NA, T >::size(), Rcpp::sugar::Minus\_↵ Primitive\_Vector< RTYPE, false, T >::size(), and Rcpp::sugar::Minus\_Primitive\_Vector< REALSXP, false, T >::size().

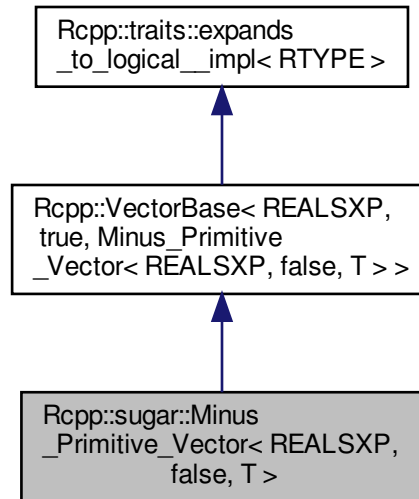
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/minus.h](#)

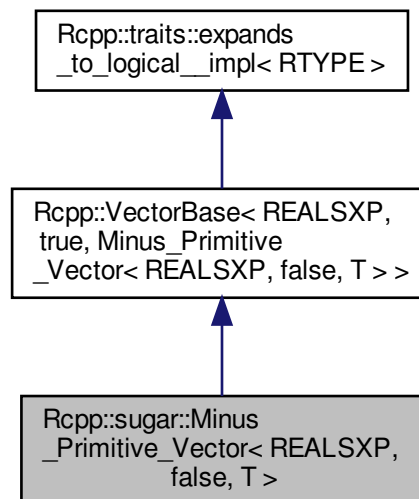
## 6.405 Rcpp::sugar::Minus\_Primitive\_Vector< REALSXP, false, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for `Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >`:



Collaboration diagram for `Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >`:





## Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, false, T > [VEC\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, T >::type [VEC\\_EXT](#)

## Public Member Functions

- [Minus\\_Primitive\\_Vector](#) (double lhs\_, const [VEC\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- double [lhs](#)
- const [VEC\\_EXT](#) & [rhs](#)

### 6.405.1 Detailed Description

```
template<typename T>
class Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >
```

Definition at line 392 of file minus.h.

### 6.405.2 Member Typedef Documentation

#### 6.405.2.1 VEC\_EXT

```
template<typename T >
typedef Rcpp::traits::Extractor<REALSXP,false,T>::type Rcpp::sugar::Minus\_Primitive\_Vector< REALSXP,
false, T >::VEC_EXT
```

Definition at line 396 of file minus.h.

#### 6.405.2.2 VEC\_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP,false,T> Rcpp::sugar::Minus\_Primitive\_Vector< REALSXP, false, T
>::VEC_TYPE
```

Definition at line 395 of file minus.h.

## 6.405.3 Constructor & Destructor Documentation

### 6.405.3.1 Minus\_Primitive\_Vector()

```
template<typename T >
Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::Minus_Primitive_Vector (
    double lhs_,
    const VEC_TYPE & rhs_ ) [inline]
```

Definition at line 398 of file minus.h.

## 6.405.4 Member Function Documentation

### 6.405.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 401 of file minus.h.

References `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs`, and `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs`.

### 6.405.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::size ( ) const [inline]
```

Definition at line 405 of file minus.h.

References `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs`.

## 6.405.5 Member Data Documentation

## 6.405.5.1 lhs

```
template<typename T >
double Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::lhs [private]
```

Definition at line 408 of file minus.h.

## 6.405.5.2 rhs

```
template<typename T >
const VEC_EXT& Rcpp::sugar::Minus_Primitive_Vector< REALSXP, false, T >::rhs [private]
```

Definition at line 409 of file minus.h.

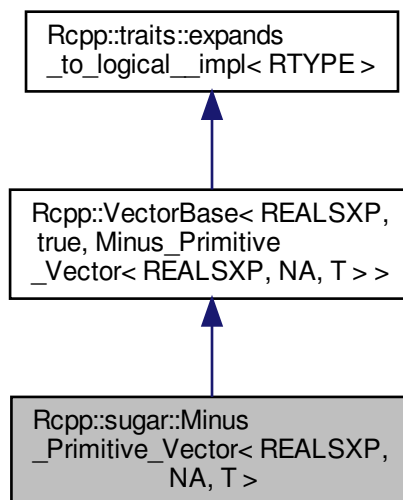
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

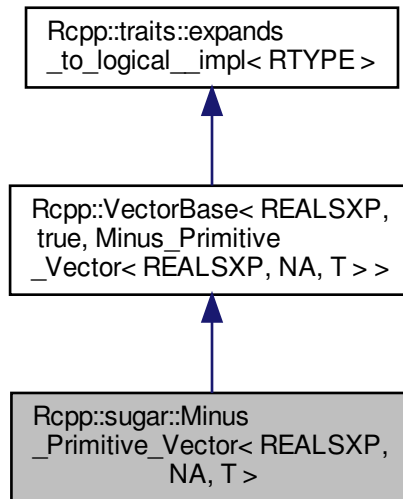
## 6.406 Rcpp::sugar::Minus\_Primitive\_Vector< REALSXP, NA, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Primitive\_Vector< REALSXP, NA, T >:



Collaboration diagram for `Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >`:



## Public Types

- typedef `Rcpp::VectorBase< REALSXP, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< REALSXP, NA, T >::type` `VEC_EXT`

## Public Member Functions

- `Minus_Primitive_Vector` (double lhs\_, const `VEC_TYPE` &rhs\_)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- double lhs
- const `VEC_EXT` & rhs

### 6.406.1 Detailed Description

```

template<bool NA, typename T>
class Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >
  
```

Definition at line 346 of file `minus.h`.

## 6.406.2 Member Typedef Documentation

### 6.406.2.1 VEC\_EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP,NA,T>::type Rcpp::sugar::Minus_Primitive_Vector< REALSXP,
NA, T >::VEC_EXT
```

Definition at line 350 of file minus.h.

### 6.406.2.2 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >↔
::VEC_TYPE
```

Definition at line 349 of file minus.h.

## 6.406.3 Constructor & Destructor Documentation

### 6.406.3.1 Minus\_Primitive\_Vector()

```
template<bool NA, typename T >
Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::Minus_Primitive_Vector (
    double lhs_,
    const VEC_TYPE & rhs_ ) [inline]
```

Definition at line 352 of file minus.h.

## 6.406.4 Member Function Documentation

#### 6.406.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::operator[] (
    R_xlen_t i) const [inline]
```

Definition at line 355 of file minus.h.

References `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs`, and `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs`.

#### 6.406.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::size ( ) const [inline]
```

Definition at line 358 of file minus.h.

References `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs`.

### 6.406.5 Member Data Documentation

#### 6.406.5.1 lhs

```
template<bool NA, typename T >
double Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::lhs [private]
```

Definition at line 361 of file minus.h.

#### 6.406.5.2 rhs

```
template<bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Minus_Primitive_Vector< REALSXP, NA, T >::rhs [private]
```

Definition at line 362 of file minus.h.

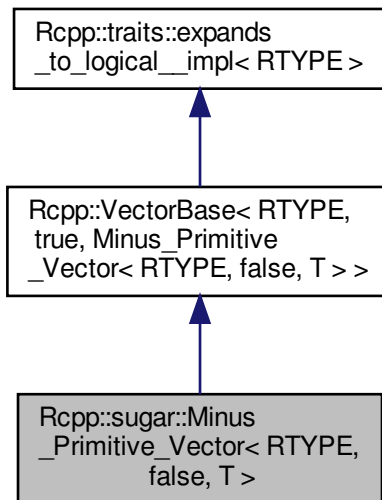
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/operators/minus.h`

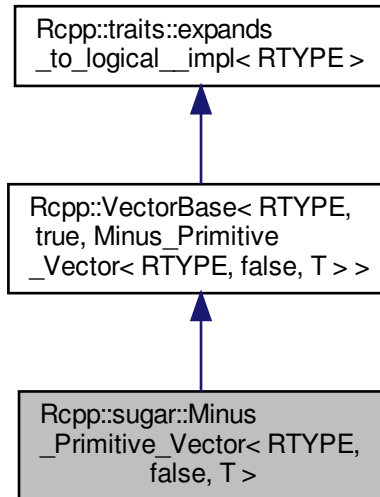
## 6.407 Rcpp::sugar::Minus\_Primitive\_Vector< RTYPE, false, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Primitive\_Vector< RTYPE, false, T >:



Collaboration diagram for `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >`:



## Public Types

- typedef `Rcpp::VectorBase< RTYPE, false, T >` `VEC_TYPE`
- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::traits::Extractor< REALSXP, false, T >::type` `VEC_EXT`

## Public Member Functions

- `Minus_Primitive_Vector` (`STORAGE lhs_`, `const VEC_TYPE &rhs_`)
- `STORAGE operator[]` (`R_xlen_t i`) `const`
- `R_xlen_t size` () `const`

## Private Attributes

- `STORAGE lhs`
- `const VEC_EXT & rhs`
- `bool lhs_na`

### 6.407.1 Detailed Description

```

template<int RTYPE, typename T>
class Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >
  
```

Definition at line 368 of file `minus.h`.



## 6.407.2 Member Typedef Documentation

### 6.407.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T
>::STORAGE
```

Definition at line 372 of file minus.h.

### 6.407.2.2 VEC\_EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor<REALSXP, false, T>::type Rcpp::sugar::Minus_Primitive_Vector< RTYPE,
false, T >::VEC_EXT
```

Definition at line 373 of file minus.h.

### 6.407.2.3 VEC\_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE, false, T> Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >↔
::VEC_TYPE
```

Definition at line 371 of file minus.h.

## 6.407.3 Constructor & Destructor Documentation

### 6.407.3.1 Minus\_Primitive\_Vector()

```
template<int RTYPE, typename T >
Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::Minus_Primitive_Vector (
    STORAGE lhs_,
    const VEC_TYPE & rhs_ ) [inline]
```

Definition at line 375 of file minus.h.

## 6.407.4 Member Function Documentation

### 6.407.4.1 operator[]()

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::operator[] (  
    R_xlen_t i ) const [inline]
```

Definition at line 378 of file minus.h.

References `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs`, `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::lhs_na`, and `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs`.

### 6.407.4.2 size()

```
template<int RTYPE, typename T >  
R_xlen_t Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::size ( ) const [inline]
```

Definition at line 383 of file minus.h.

References `Rcpp::sugar::Minus_Primitive_Vector< RTYPE, NA, T >::rhs`.

## 6.407.5 Member Data Documentation

### 6.407.5.1 lhs

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::lhs [private]
```

Definition at line 386 of file minus.h.

### 6.407.5.2 lhs\_na

```
template<int RTYPE, typename T >  
bool Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::lhs_na [private]
```

Definition at line 388 of file minus.h.

### 6.407.5.3 rhs

```
template<int RTYPE, typename T >  
const VEC_EXT& Rcpp::sugar::Minus_Primitive_Vector< RTYPE, false, T >::rhs [private]
```

Definition at line 387 of file minus.h.

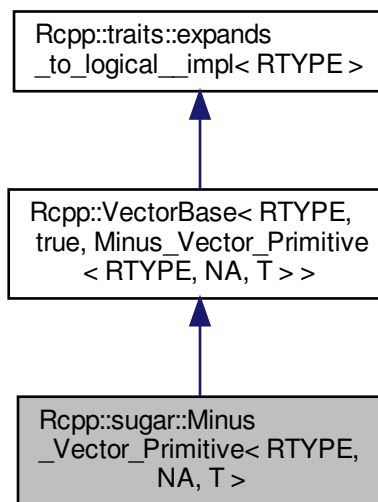
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

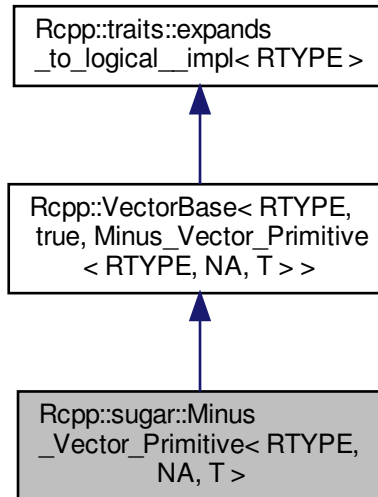
## 6.408 Rcpp::sugar::Minus\_Vector\_Primitive< RTYPE, NA, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Vector\_Primitive< RTYPE, NA, T >:



Collaboration diagram for `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >`:



## Public Types

- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< RTYPE, NA, T >::type` `VEC_EXT`

## Public Member Functions

- `Minus_Vector_Primitive` (const `VEC_TYPE` &lhs\_, `STORAGE` rhs\_)
- `STORAGE operator[]` (R\_xlen\_t i) const
- `R_xlen_t size` () const

## Private Attributes

- const `VEC_EXT` & lhs
- `STORAGE` rhs
- bool rhs\_na

### 6.408.1 Detailed Description

```

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >
  
```

Definition at line 228 of file `minus.h`.

## 6.408.2 Member Typedef Documentation

### 6.408.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >↔
::STORAGE
```

Definition at line 231 of file minus.h.

### 6.408.2.2 VEC\_EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor<RTYPE,NA,T>::type Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA,
T >::VEC_EXT
```

Definition at line 233 of file minus.h.

### 6.408.2.3 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 232 of file minus.h.

## 6.408.3 Constructor & Destructor Documentation

### 6.408.3.1 Minus\_Vector\_Primitive()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::Minus_Vector_Primitive (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 235 of file minus.h.

## 6.408.4 Member Function Documentation

### 6.408.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 238 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs`, `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs`, and `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs_na`.

### 6.408.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 244 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs`.

## 6.408.5 Member Data Documentation

### 6.408.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs [private]
```

Definition at line 247 of file minus.h.

Referenced by `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::operator[]()`, `Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::operator[]()`, `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::operator[]()`, `Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::operator[]()`, `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::size()`, `Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::size()`, `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::size()`, and `Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::size()`.

### 6.408.5.2 rhs

```
template<int RTYPE, bool NA, typename T >  
STORAGE Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs [private]
```

Definition at line 248 of file minus.h.

Referenced by Rcpp::sugar::Minus\_Vector\_Primitive< RTYPE, NA, T >::operator[](), Rcpp::sugar::Minus\_Vector\_↔ Primitive< REALSXP, NA, T >::operator[](), Rcpp::sugar::Minus\_Vector\_Primitive< RTYPE, false, T >::operator[](), and Rcpp::sugar::Minus\_Vector\_Primitive< REALSXP, false, T >::operator[]().

### 6.408.5.3 rhs\_na

```
template<int RTYPE, bool NA, typename T >  
bool Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs_na [private]
```

Definition at line 249 of file minus.h.

Referenced by Rcpp::sugar::Minus\_Vector\_Primitive< RTYPE, NA, T >::operator[](), and Rcpp::sugar::Minus\_Vector\_↔ \_Primitive< RTYPE, false, T >::operator[]().

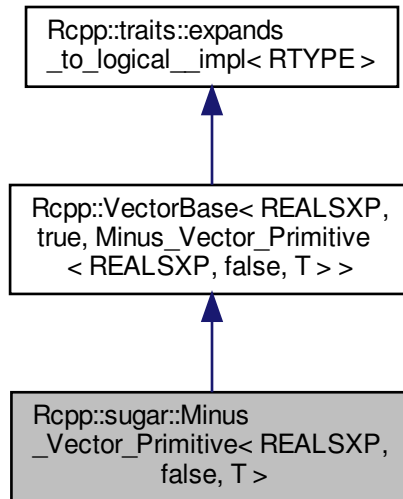
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/minus.h](#)

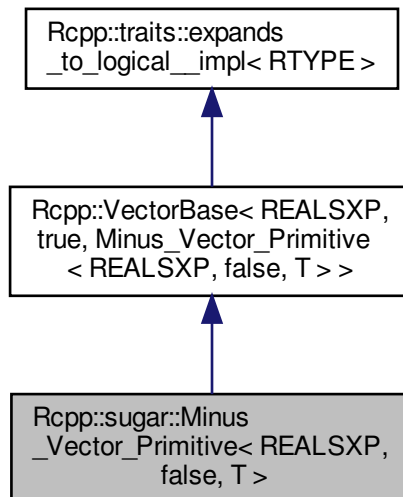
## 6.409 Rcpp::sugar::Minus\_Vector\_Primitive< REALSXP, false, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for `Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >`:



Collaboration diagram for `Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >`:





## Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, false, T > [VEC\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, T >::type [VEC\\_EXT](#)

## Public Member Functions

- [Minus\\_Vector\\_Primitive](#) (const [VEC\\_TYPE](#) &lhs\_, double rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [VEC\\_EXT](#) & lhs
- double rhs

### 6.409.1 Detailed Description

```
template<typename T>
class Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >
```

Definition at line 298 of file minus.h.

### 6.409.2 Member Typedef Documentation

#### 6.409.2.1 VEC\_EXT

```
template<typename T >
typedef Rcpp::traits::Extractor<REALSXP,false,T>::type Rcpp::sugar::Minus\_Vector\_Primitive< REALSXP,
false, T >::VEC_EXT
```

Definition at line 302 of file minus.h.

#### 6.409.2.2 VEC\_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP,false,T> Rcpp::sugar::Minus\_Vector\_Primitive< REALSXP, false, T
>::VEC_TYPE
```

Definition at line 301 of file minus.h.

## 6.409.3 Constructor & Destructor Documentation

### 6.409.3.1 Minus\_Vector\_Primitive()

```
template<typename T >
Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::Minus_Vector_Primitive (
    const VEC_TYPE & lhs_,
    double rhs_ ) [inline]
```

Definition at line 304 of file minus.h.

## 6.409.4 Member Function Documentation

### 6.409.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 307 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs`, and `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs`.

### 6.409.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::size ( ) const [inline]
```

Definition at line 311 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs`.

## 6.409.5 Member Data Documentation

## 6.409.5.1 lhs

```
template<typename T >
const VEC_EXT& Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::lhs [private]
```

Definition at line 314 of file minus.h.

## 6.409.5.2 rhs

```
template<typename T >
double Rcpp::sugar::Minus_Vector_Primitive< REALSXP, false, T >::rhs [private]
```

Definition at line 315 of file minus.h.

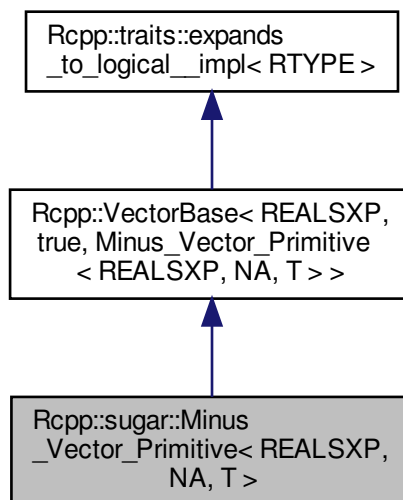
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

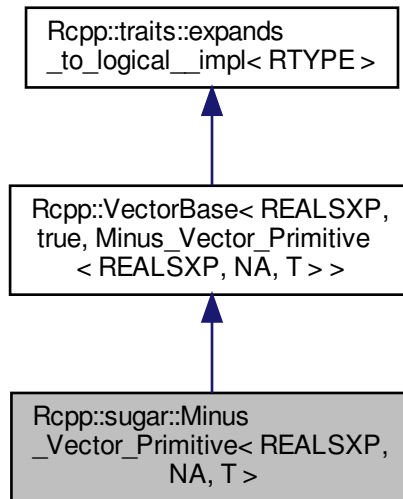
## 6.410 Rcpp::sugar::Minus\_Vector\_Primitive< REALSXP, NA, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Vector\_Primitive< REALSXP, NA, T >:



Collaboration diagram for `Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >`:



## Public Types

- typedef `Rcpp::VectorBase< REALSXP, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< REALSXP, NA, T >::type` `VEC_EXT`

## Public Member Functions

- `Minus_Vector_Primitive` (const `VEC_TYPE` &lhs\_, double rhs\_)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const `VEC_EXT` & lhs
- double rhs

### 6.410.1 Detailed Description

```

template<bool NA, typename T>
class Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >
  
```

Definition at line 252 of file `minus.h`.

## 6.410.2 Member Typedef Documentation

### 6.410.2.1 VEC\_EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP,NA,T>::type Rcpp::sugar::Minus_Vector_Primitive< REALSXP,
NA, T >::VEC_EXT
```

Definition at line 256 of file minus.h.

### 6.410.2.2 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >↔
::VEC_TYPE
```

Definition at line 255 of file minus.h.

## 6.410.3 Constructor & Destructor Documentation

### 6.410.3.1 Minus\_Vector\_Primitive()

```
template<bool NA, typename T >
Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::Minus_Vector_Primitive (
    const VEC_TYPE & lhs_,
    double rhs_ ) [inline]
```

Definition at line 258 of file minus.h.

## 6.410.4 Member Function Documentation

#### 6.410.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::operator[] (
    R_xlen_t i) const [inline]
```

Definition at line 261 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs`, and `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs`.

#### 6.410.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::size ( ) const [inline]
```

Definition at line 265 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs`.

### 6.410.5 Member Data Documentation

#### 6.410.5.1 lhs

```
template<bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::lhs [private]
```

Definition at line 268 of file minus.h.

#### 6.410.5.2 rhs

```
template<bool NA, typename T >
double Rcpp::sugar::Minus_Vector_Primitive< REALSXP, NA, T >::rhs [private]
```

Definition at line 269 of file minus.h.

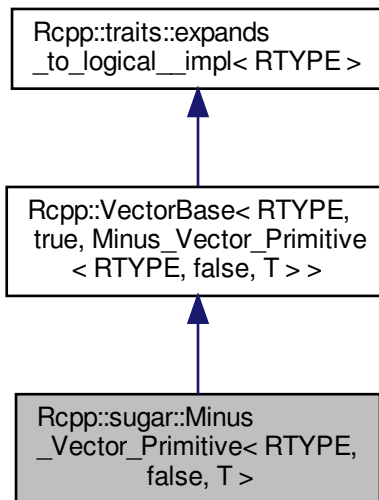
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/operators/minus.h`

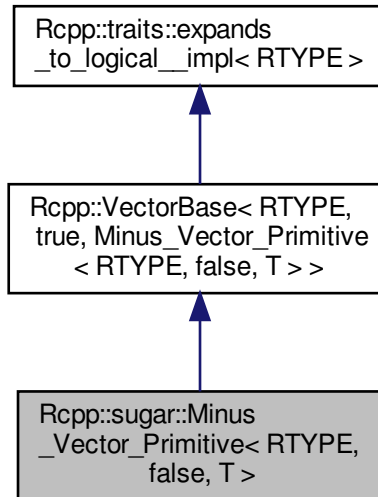
## 6.411 Rcpp::sugar::Minus\_Vector\_Primitive< RTYPE, false, T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Vector\_Primitive< RTYPE, false, T >:



Collaboration diagram for `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >`:



## Public Types

- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::VectorBase< RTYPE, false, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< RTYPE, false, T >::type` `VEC_EXT`

## Public Member Functions

- `Minus_Vector_Primitive` (const `VEC_TYPE` &lhs\_, `STORAGE` rhs\_)
- `STORAGE operator[]` (R\_xlen\_t i) const
- `R_xlen_t size` () const

## Private Attributes

- const `VEC_EXT` & lhs
- `STORAGE` rhs
- bool rhs\_na

### 6.411.1 Detailed Description

```

template<int RTYPE, typename T>
class Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >
  
```

Definition at line 274 of file `minus.h`.



## 6.411.2 Member Typedef Documentation

### 6.411.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T
>::STORAGE
```

Definition at line 277 of file minus.h.

### 6.411.2.2 VEC\_EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor<RTYPE, false, T>::type Rcpp::sugar::Minus_Vector_Primitive< RTYPE,
false, T >::VEC_EXT
```

Definition at line 279 of file minus.h.

### 6.411.2.3 VEC\_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE, false, T> Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >↔
::VEC_TYPE
```

Definition at line 278 of file minus.h.

## 6.411.3 Constructor & Destructor Documentation

### 6.411.3.1 Minus\_Vector\_Primitive()

```
template<int RTYPE, typename T >
Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::Minus_Vector_Primitive (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 281 of file minus.h.

## 6.411.4 Member Function Documentation

### 6.411.4.1 operator[]()

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::operator[] (  
    R_xlen_t i ) const [inline]
```

Definition at line 284 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs`, `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs`, and `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::rhs_na`.

### 6.411.4.2 size()

```
template<int RTYPE, typename T >  
R_xlen_t Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::size ( ) const [inline]
```

Definition at line 290 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Primitive< RTYPE, NA, T >::lhs`.

## 6.411.5 Member Data Documentation

### 6.411.5.1 lhs

```
template<int RTYPE, typename T >  
const VEC_EXT& Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::lhs [private]
```

Definition at line 293 of file minus.h.

### 6.411.5.2 rhs

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::rhs [private]
```

Definition at line 294 of file minus.h.

### 6.411.5.3 rhs\_na

```
template<int RTYPE, typename T >
bool Rcpp::sugar::Minus_Vector_Primitive< RTYPE, false, T >::rhs_na [private]
```

Definition at line 295 of file minus.h.

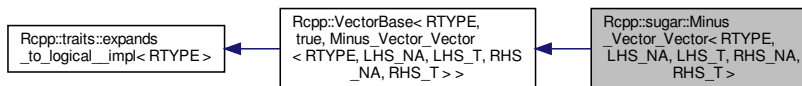
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

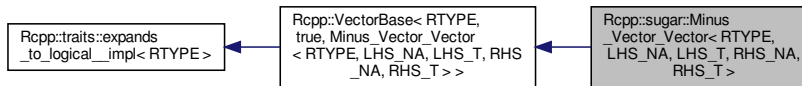
## 6.412 Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase< RTYPE, LHS\\_NA, LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase< RTYPE, RHS\\_NA, RHS\\_T >](#) [RHS\\_TYPE](#)
- typedef [traits::storage\\_type< RTYPE >::type](#) [STORAGE](#)
- typedef [Rcpp::traits::Extractor< RTYPE, LHS\\_NA, LHS\\_T >::type](#) [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor< RTYPE, RHS\\_NA, RHS\\_T >::type](#) [RHS\\_EXT](#)

## Public Member Functions

- [Minus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) ([R\\_xlen\\_t](#) i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & lhs
- const [RHS\\_EXT](#) & rhs

### 6.412.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 29 of file minus.h.

### 6.412.2 Member Typedef Documentation

#### 6.412.2.1 LHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 35 of file minus.h.

#### 6.412.2.2 LHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE, LHS_NA, LHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 32 of file minus.h.

### 6.412.2.3 RHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 36 of file minus.h.

### 6.412.2.4 RHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 33 of file minus.h.

### 6.412.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 34 of file minus.h.

## 6.412.3 Constructor & Destructor Documentation

### 6.412.3.1 Minus\_Vector\_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Minus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 38 of file minus.h.

### 6.412.4 Member Function Documentation

### 6.412.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 41 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

### 6.412.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 48 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

## 6.412.5 Member Data Documentation

### 6.412.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 51 of file minus.h.

Referenced by `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[]()`, `Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[]()`, `Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[]()`, `Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[]()`, `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[]()`, `Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[]()`, `Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[]()`, `Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[]()`, `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size()`, `Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size()`, `Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size()`, `Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size()`, `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size()`, `Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size()`, and `Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size()`.

## 6.413 Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T > Class Template Reference

### 6.412.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs [private]
```

Definition at line 52 of file minus.h.

Referenced by Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >::operator[](), Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >::operator[](), Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >::operator[](), and Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >::operator[]().

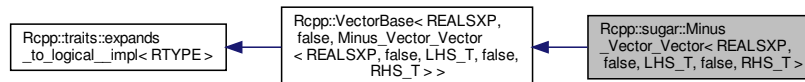
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

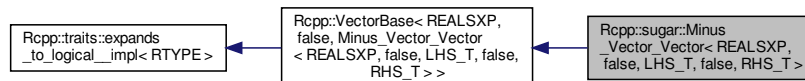
## 6.413 Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< REALSXP, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, RHS\_T >::type [RHS\\_EXT](#)

## Public Member Functions

- [Minus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) ([R\\_xlen\\_t](#) i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & lhs
- const [RHS\\_EXT](#) & rhs

### 6.413.1 Detailed Description

```
template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
```

Definition at line 201 of file minus.h.

### 6.413.2 Member Typedef Documentation

#### 6.413.2.1 LHS\_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
REALSXP, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 206 of file minus.h.

#### 6.413.2.2 LHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, false, LHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 204 of file minus.h.



### 6.413.2.3 RHS\_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
REALSXP, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 207 of file minus.h.

### 6.413.2.4 RHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, false, RHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 205 of file minus.h.

## 6.413.3 Constructor & Destructor Documentation

### 6.413.3.1 Minus\_Vector\_Vector()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::Minus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 209 of file minus.h.

## 6.413.4 Member Function Documentation

### 6.413.4.1 operator[]()

```
template<typename LHS_T , typename RHS_T >
double Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 212 of file minus.h.

References [Rcpp::sugar::Minus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#), and [Rcpp::sugar::Minus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::rhs](#).

#### 6.413.4.2 size()

```
template<typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 216 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

### 6.413.5 Member Data Documentation

#### 6.413.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::lhs [private]
```

Definition at line 219 of file minus.h.

#### 6.413.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::rhs [private]
```

Definition at line 220 of file minus.h.

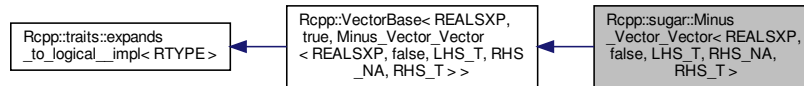
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/minus.h](#)

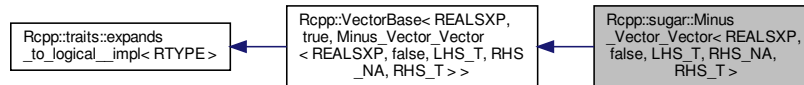
## 6.414 Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase< REALSXP, false, LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase< REALSXP, RHS\\_NA, RHS\\_T >](#) [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor< REALSXP, false, LHS\\_T >::type](#) [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor< REALSXP, RHS\\_NA, RHS\\_T >::type](#) [RHS\\_EXT](#)

### Public Member Functions

- [Minus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

### 6.414.1 Detailed Description

```
template<typename LHS_T, bool RHS_NA, typename RHS_T>  
class Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
```

Definition at line 105 of file minus.h.

### 6.414.2 Member Typedef Documentation

#### 6.414.2.1 LHS\_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<  
REALSXP, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 110 of file minus.h.

#### 6.414.2.2 LHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::VectorBase<REALSXP, false, LHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, false,  
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 108 of file minus.h.

#### 6.414.2.3 RHS\_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<  
REALSXP, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 111 of file minus.h.

#### 6.414.2.4 RHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,RHS_NA,RHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 109 of file minus.h.

### 6.414.3 Constructor & Destructor Documentation

#### 6.414.3.1 Minus\_Vector\_Vector()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::Minus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 113 of file minus.h.

### 6.414.4 Member Function Documentation

#### 6.414.4.1 operator[]()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 116 of file minus.h.

References Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

#### 6.414.4.2 size()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 120 of file minus.h.

References Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs.

## 6.414.5 Member Data Documentation

### 6.414.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 123 of file minus.h.

### 6.414.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 124 of file minus.h.

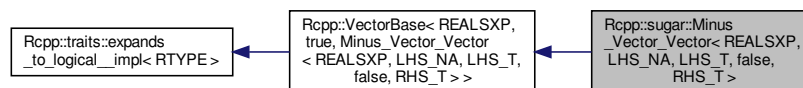
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/minus.h](#)

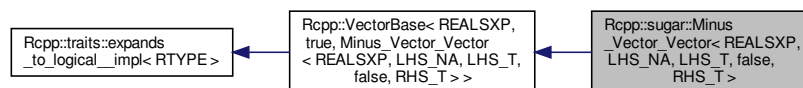
## 6.415 Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< REALSXP, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, RHS\_T >::type [RHS\\_EXT](#)

## Public Member Functions

- [Minus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & lhs
- const [RHS\\_EXT](#) & rhs

### 6.415.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
```

Definition at line 154 of file minus.h.

### 6.415.2 Member Typedef Documentation

#### 6.415.2.1 LHS\_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Minus\_Vector\_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 159 of file minus.h.

### 6.415.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,LHS_NA,LHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 157 of file minus.h.

### 6.415.2.3 RHS\_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 160 of file minus.h.

### 6.415.2.4 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,false,RHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 158 of file minus.h.

## 6.415.3 Constructor & Destructor Documentation

### 6.415.3.1 Minus\_Vector\_Vector()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::Minus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 162 of file minus.h.

### 6.415.4 Member Function Documentation



### 6.415.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
double Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 165 of file minus.h.

References Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.415.4.2 size()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 169 of file minus.h.

References Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs.

## 6.415.5 Member Data Documentation

### 6.415.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::lhs
[private]
```

Definition at line 172 of file minus.h.

### 6.415.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::rhs
[private]
```

Definition at line 173 of file minus.h.

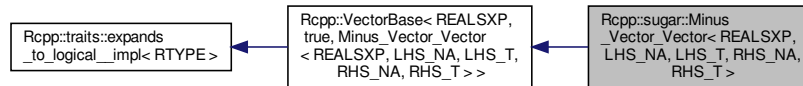
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

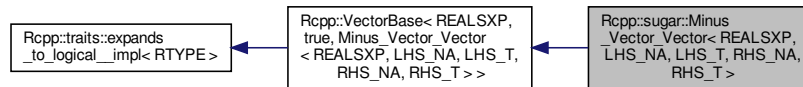
## 6.416 Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Minus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< REALSXP, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, RHS\_NA, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Minus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

## 6.416.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>  
class Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 56 of file minus.h.

## 6.416.2 Member Typedef Documentation

### 6.416.2.1 LHS\_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<  
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 61 of file minus.h.

### 6.416.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::VectorBase<REALSXP, LHS_NA, LHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA,  
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 59 of file minus.h.

### 6.416.2.3 RHS\_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<  
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 62 of file minus.h.

### 6.416.2.4 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,RHS_NA,RHS_T> Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 60 of file minus.h.

## 6.416.3 Constructor & Destructor Documentation

### 6.416.3.1 Minus\_Vector\_Vector()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::Minus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 64 of file minus.h.

## 6.416.4 Member Function Documentation

### 6.416.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 67 of file minus.h.

References [Rcpp::sugar::Minus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#), and [Rcpp::sugar::Minus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::rhs](#).

### 6.416.4.2 size()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( )
const [inline]
```

Definition at line 71 of file minus.h.

References [Rcpp::sugar::Minus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#).

## 6.416.5 Member Data Documentation

### 6.416.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 74 of file minus.h.

### 6.416.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 75 of file minus.h.

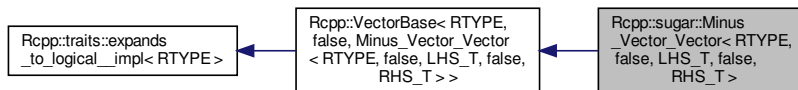
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

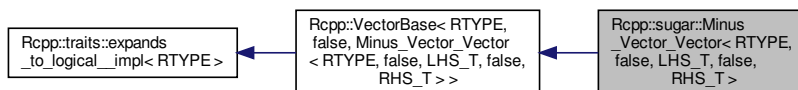
## 6.417 Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, RHS\_T >::type [RHS\\_EXT](#)

## Public Member Functions

- [Minus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & lhs
- const [RHS\\_EXT](#) & rhs

### 6.417.1 Detailed Description

```
template<int RTYPE, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
```

Definition at line 178 of file minus.h.

### 6.417.2 Member Typedef Documentation

#### 6.417.2.1 LHS\_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, false, LHS_T>::type Rcpp::sugar::Minus\_Vector\_Vector<
RTYPE, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 184 of file minus.h.

### 6.417.2.2 LHS\_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 181 of file minus.h.

### 6.417.2.3 RHS\_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, false, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 185 of file minus.h.

### 6.417.2.4 RHS\_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 182 of file minus.h.

### 6.417.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::STORAGE
```

Definition at line 183 of file minus.h.

## 6.417.3 Constructor & Destructor Documentation

### 6.417.3.1 Minus\_Vector\_Vector()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::Minus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 187 of file minus.h.

## 6.417.4 Member Function Documentation

### 6.417.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 190 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

### 6.417.4.2 size()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 194 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

## 6.417.5 Member Data Documentation

### 6.417.5.1 lhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::lhs [private]
```

Definition at line 197 of file minus.h.



6.417.5.2 rhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::rhs [private]
```

Definition at line 198 of file minus.h.

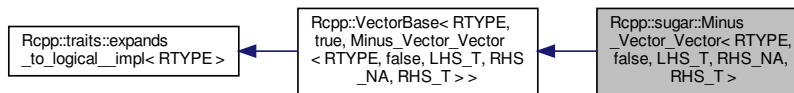
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/minus.h

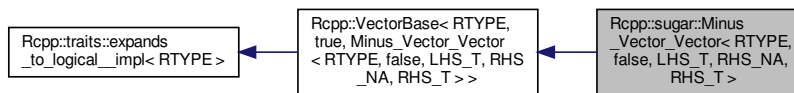
6.418 Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >:



Public Types

- typedef [Rcpp::VectorBase< RTYPE, false, LHS\\_T >](#) LHS\_TYPE
- typedef [Rcpp::VectorBase< RTYPE, RHS\\_NA, RHS\\_T >](#) RHS\_TYPE
- typedef [traits::storage\\_type< RTYPE >::type](#) STORAGE
- typedef [Rcpp::traits::Extractor< RTYPE, false, LHS\\_T >::type](#) LHS\_EXT
- typedef [Rcpp::traits::Extractor< RTYPE, RHS\\_NA, RHS\\_T >::type](#) RHS\_EXT

## Public Member Functions

- [Minus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) ([R\\_xlen\\_t](#) i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & lhs
- const [RHS\\_EXT](#) & rhs

### 6.418.1 Detailed Description

```
template<int RTYPE, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
```

Definition at line 80 of file minus.h.

### 6.418.2 Member Typedef Documentation

#### 6.418.2.1 LHS\_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, false, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 85 of file minus.h.

#### 6.418.2.2 LHS\_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE, false, LHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS←
_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 82 of file minus.h.

### 6.418.2.3 RHS\_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 86 of file minus.h.

### 6.418.2.4 RHS\_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 83 of file minus.h.

### 6.418.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 84 of file minus.h.

## 6.418.3 Constructor & Destructor Documentation

### 6.418.3.1 Minus\_Vector\_Vector()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::Minus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 88 of file minus.h.

### 6.418.4 Member Function Documentation

#### 6.418.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >  
STORAGE Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[] (   
    R_xlen_t i ) const [inline]
```

Definition at line 91 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

#### 6.418.4.2 size()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >  
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size ( ) const  
[inline]
```

Definition at line 97 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

### 6.418.5 Member Data Documentation

#### 6.418.5.1 lhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >  
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 100 of file minus.h.

#### 6.418.5.2 rhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >  
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::rhs [private]
```

Definition at line 101 of file minus.h.

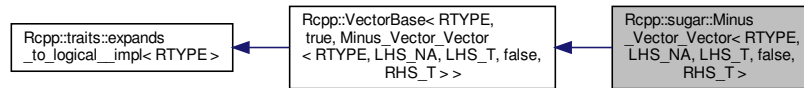
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/operators/minus.h`

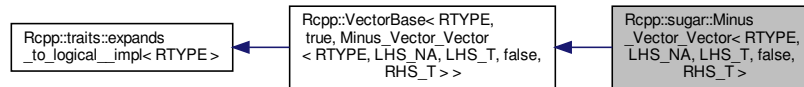
## 6.419 Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <minus.h>
```

Inheritance diagram for Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Minus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

### 6.419.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, typename RHS_T>  
class Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
```

Definition at line 129 of file minus.h.

### 6.419.2 Member Typedef Documentation

#### 6.419.2.1 LHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef Rcpp::traits::Extractor<RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Minus_Vector_Vector<  
RTYPE, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 134 of file minus.h.

#### 6.419.2.2 LHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef Rcpp::VectorBase<RTYPE, LHS_NA, LHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA,  
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 131 of file minus.h.

#### 6.419.2.3 RHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef Rcpp::traits::Extractor<RTYPE, false, RHS_T>::type Rcpp::sugar::Minus_Vector_Vector<  
RTYPE, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 135 of file minus.h.

## 6.419 Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T > Class Template Reference

### 6.419.2.4 RHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 132 of file minus.h.

### 6.419.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T,
false, RHS_T >::STORAGE
```

Definition at line 133 of file minus.h.

## 6.419.3 Constructor & Destructor Documentation

### 6.419.3.1 Minus\_Vector\_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::Minus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 137 of file minus.h.

## 6.419.4 Member Function Documentation

### 6.419.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 140 of file minus.h.

References Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Minus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.419.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 146 of file minus.h.

References `Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

## 6.419.5 Member Data Documentation

### 6.419.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::lhs [private]
```

Definition at line 149 of file minus.h.

### 6.419.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::rhs [private]
```

Definition at line 150 of file minus.h.

The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/operators/minus.h`

## 6.420 Rcpp::Module Class Reference

```
#include <Module.h>
```

### Public Types

- typedef `std::map< std::string, CppFunction * >` `MAP`
- typedef `std::pair< const std::string, CppFunction * >` `FUNCTION_PAIR`
- typedef `std::map< std::string, class_Base * >` `CLASS_MAP`
- typedef `std::pair< const std::string, class_Base * >` `CLASS_PAIR`
- typedef `CLASS_MAP::iterator` `CLASS_ITERATOR`



## Public Member Functions

- [Module](#) ()
- [Module](#) (const char \*name\_)
- [SEXP invoke](#) (const std::string &name\_, SEXP \*args, int nargs)
- [IntegerVector functions\\_arity](#) ()
- [CharacterVector functions\\_names](#) ()
- [CharacterVector class\\_names](#) ()
- [List classes\\_info](#) ()
- [CharacterVector complete](#) ()
- [SEXP get\\_function](#) (const std::string &name\_)
- [DL\\_FUNC get\\_function\\_ptr](#) (const std::string &name\_)
- void [Add](#) (const char \*name\_, [CppClass](#) \*ptr)
- void [AddClass](#) (const char \*name\_, [class\\_Base](#) \*cptr)
- bool [has\\_function](#) (const std::string &m)
- bool [has\\_class](#) (const std::string &m)
- [CppClass](#) [get\\_class](#) (const std::string &cl)
- [class\\_Base](#) \* [get\\_class\\_pointer](#) (const std::string &cl)
- void [add\\_enum](#) (const std::string &parent\_class\_typeinfo\_name, const std::string &enum\_name, const std::map<std::string, int > &value)

## Public Attributes

- std::string [name](#)

## Private Attributes

- [MAP](#) functions
- [CLASS\\_MAP](#) classes
- std::string [prefix](#)

### 6.420.1 Detailed Description

holds information about exposed functions and classes

Definition at line 29 of file Module.h.

### 6.420.2 Member Typedef Documentation

#### 6.420.2.1 CLASS\_ITERATOR

```
typedef CLASS_MAP::iterator Rcpp::Module::CLASS_ITERATOR
```

Definition at line 36 of file Module.h.

### 6.420.2.2 CLASS\_MAP

```
typedef std::map<std::string, class_Base*> Rcpp::Module::CLASS_MAP
```

Definition at line 34 of file Module.h.

### 6.420.2.3 CLASS\_PAIR

```
typedef std::pair<const std::string, class_Base*> Rcpp::Module::CLASS_PAIR
```

Definition at line 35 of file Module.h.

### 6.420.2.4 FUNCTION\_PAIR

```
typedef std::pair<const std::string, CppFunction*> Rcpp::Module::FUNCTION_PAIR
```

Definition at line 32 of file Module.h.

### 6.420.2.5 MAP

```
typedef std::map<std::string, CppFunction*> Rcpp::Module::MAP
```

Definition at line 31 of file Module.h.

## 6.420.3 Constructor & Destructor Documentation

### 6.420.3.1 Module() [1/2]

```
Rcpp::Module::Module ( ) [inline]
```

Definition at line 38 of file Module.h.

### 6.420.3.2 Module() [2/2]

```
Rcpp::Module::Module (  
    const char * name_ ) [inline]
```

Definition at line 41 of file Module.h.

References name, and prefix.

## 6.420.4 Member Function Documentation

### 6.420.4.1 Add()

```
void Rcpp::Module::Add (  
    const char * name_,  
    CppFunction * ptr ) [inline]
```

Definition at line 192 of file Module.h.

References functions, Rcpp::CppFunction::get\_function\_ptr(), and prefix.

Here is the call graph for this function:



#### 6.420.4.2 add\_enum()

```
void Rcpp::Module::add_enum (
    const std::string & parent_class_typeinfo_name,
    const std::string & enum_name,
    const std::map< std::string, int > & value ) [inline]
```

Definition at line 219 of file Module.h.

References Rcpp::class\_Base::add\_enum(), and classes.

Referenced by Rcpp::enum\_< Enum, Parent >::~~enum\_().

Here is the call graph for this function:



#### 6.420.4.3 AddClass()

```
void Rcpp::Module::AddClass (
    const char * name_,
    class_Base * cptr ) [inline]
```

Definition at line 197 of file Module.h.

References classes.

#### 6.420.4.4 class\_names()

```
CharacterVector Rcpp::Module::class_names ( ) [inline]
```

exposed class names

Definition at line 101 of file Module.h.

References classes.

#### 6.420.4.5 classes\_info()

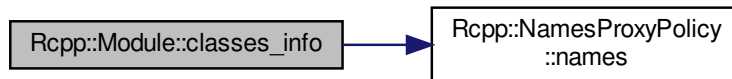
`List` Rcpp::Module::classes\_info ( ) [inline]

information about the classes

Definition at line 25 of file Module.h.

References classes, and Rcpp::NamesProxyPolicy< CLASS >::names().

Here is the call graph for this function:



#### 6.420.4.6 complete()

`CharacterVector` Rcpp::Module::complete ( ) [inline]

completion information

Definition at line 119 of file Module.h.

References classes, and functions.

#### 6.420.4.7 functions\_arity()

`IntegerVector` Rcpp::Module::functions\_arity ( ) [inline]

vector of arity of all the functions exported by the module

Definition at line 72 of file Module.h.

References functions, and Rcpp::NamesProxyPolicy< CLASS >::names().

Here is the call graph for this function:



#### 6.420.4.8 functions\_names()

```
CharacterVector Rcpp::Module::functions_names ( ) [inline]
```

vector of names of the functions

Definition at line 88 of file Module.h.

References functions.

#### 6.420.4.9 get\_class()

```
CppClass Rcpp::Module::get_class (
    const std::string & cl ) [inline]
```

Definition at line 39 of file Module.h.

References BEGIN\_RCPP, classes, and END\_RCPP.

#### 6.420.4.10 get\_class\_pointer()

```
class_Base* Rcpp::Module::get_class_pointer (
    const std::string & cl ) [inline]
```

Definition at line 211 of file Module.h.

References classes.

#### 6.420.4.11 get\_function()

```
SEXP Rcpp::Module::get_function (
    const std::string & name_ ) [inline]
```

Returns a list that contains:

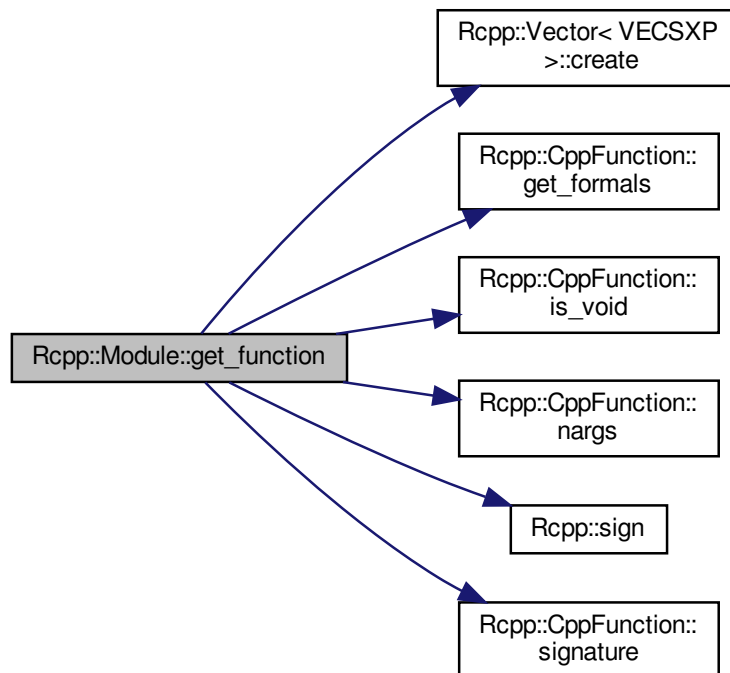
- an external pointer that encapsulates a CppFunction\*
- voidness of the function (logical)
- docstring (character)
- signature (character)
- formal arguments of the function

The [R](#) code in Module.R uses this information to create a C++Function object

Definition at line 154 of file Module.h.

References `Rcpp::Vector< VECSXP >::create()`, `Rcpp::CppFunction::docstring`, `functions`, `Rcpp::CppFunction< >::get_formals()`, `Rcpp::CppFunction::is_void()`, `Rcpp::CppFunction::nargs()`, `Rcpp::sign()`, and `Rcpp::CppFunction< >::signature()`.

Here is the call graph for this function:



#### 6.420.4.12 get\_function\_ptr()

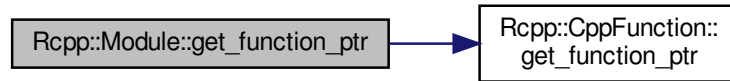
```
DL_FUNC Rcpp::Module::get_function_ptr (
    const std::string & name_ ) [inline]
```

get the underlying C++ function pointer as a DL\_FUNC

Definition at line 179 of file Module.h.

References `functions`, and `Rcpp::CppFunction::get_function_ptr()`.

Here is the call graph for this function:



#### 6.420.4.13 has\_class()

```
bool Rcpp::Module::has_class (
    const std::string & m ) [inline]
```

Definition at line 205 of file Module.h.

References classes.

#### 6.420.4.14 has\_function()

```
bool Rcpp::Module::has_function (
    const std::string & m ) [inline]
```

Definition at line 201 of file Module.h.

References functions.

#### 6.420.4.15 invoke()

```
SEXP Rcpp::Module::invoke (
    const std::string & name_,
    SEXP * args,
    int nargs ) [inline]
```

calls a function from that module with the specified arguments



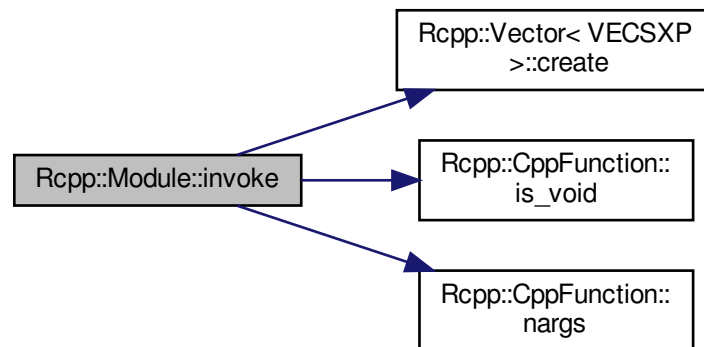
## Parameters

<i>name</i>	the name of the function to call
<i>args</i>	an array of R objects to use as arguments for the function
<i>nargs</i>	number of arguments

Definition at line 53 of file Module.h.

References `Rcpp::_`, `Rcpp::Vector< VECSXP >::create()`, `functions`, `Rcpp::CppMethod::is_void()`, and `Rcpp::CppMethod::nargs()`.

Here is the call graph for this function:



## 6.420.5 Member Data Documentation

### 6.420.5.1 classes

`CLASS_MAP` `Rcpp::Module::classes` [private]

Definition at line 235 of file Module.h.

Referenced by `add_enum()`, `AddClass()`, `class_names()`, `classes_info()`, `complete()`, `get_class()`, `get_class_pointer()`, and `has_class()`.

### 6.420.5.2 functions

```
MAP Rcpp::Module::functions [private]
```

Definition at line 234 of file Module.h.

Referenced by Add(), complete(), functions\_arity(), functions\_names(), get\_function(), get\_function\_ptr(), has\_↔function(), and invoke().

### 6.420.5.3 name

```
std::string Rcpp::Module::name
```

Definition at line 217 of file Module.h.

Referenced by Module().

### 6.420.5.4 prefix

```
std::string Rcpp::Module::prefix [private]
```

Definition at line 236 of file Module.h.

Referenced by Add(), and Module().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/module/Module.h](#)

## 6.421 Rcpp::traits::module\_wrap\_traits< T > Struct Template Reference

```
#include <module_wrap_traits.h>
```

### Public Types

- typedef [normal\\_wrap\\_tag](#) category

### 6.421.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::module_wrap_traits< T >
```

Definition at line 33 of file module\_wrap\_traits.h.

### 6.421.2 Member Typedef Documentation

#### 6.421.2.1 category

```
template<typename T >
typedef normal\_wrap\_tag Rcpp::traits::module_wrap_traits< T >::category
```

Definition at line 33 of file module\_wrap\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/module\\_wrap\\_traits.h](#)

## 6.422 Rcpp::traits::module\_wrap\_traits< T \* > Struct Template Reference

```
#include <module_wrap_traits.h>
```

### Public Types

- typedef [pointer\\_wrap\\_tag](#) category

### 6.422.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::module_wrap_traits< T * >
```

Definition at line 35 of file module\_wrap\_traits.h.

### 6.422.2 Member Typedef Documentation

### 6.422.2.1 category

```
template<typename T >
typedef pointer\_wrap\_tag Rcpp::traits::module_wrap_traits< T * >::category
```

Definition at line 35 of file `module_wrap_traits.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/module\\_wrap\\_traits.h](#)

## 6.423 Rcpp::traits::module\_wrap\_traits< void > Struct Reference

```
#include <module_wrap_traits.h>
```

### Public Types

- typedef [void\\_wrap\\_tag](#) category

### 6.423.1 Detailed Description

Definition at line 34 of file `module_wrap_traits.h`.

### 6.423.2 Member Typedef Documentation

#### 6.423.2.1 category

```
typedef void\_wrap\_tag Rcpp::traits::module_wrap_traits< void >::category
```

Definition at line 34 of file `module_wrap_traits.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/module\\_wrap\\_traits.h](#)

## 6.424 Rcpp::Na\_Proxy Class Reference

```
#include <Na_Proxy.h>
```

## Friends

- bool `operator==` (double x, [Rcpp::Na\\_Proxy](#))
- bool `operator==` (int x, [Rcpp::Na\\_Proxy](#))
- bool `operator==` ([Rcpp::String](#) x, [Rcpp::Na\\_Proxy](#))
- bool `operator==` ([Rcomplex](#) x, [Rcpp::Na\\_Proxy](#))
- bool `operator==` (SEXP x, [Rcpp::Na\\_Proxy](#))
- bool `operator==` (std::string, [Rcpp::Na\\_Proxy](#))
- bool `operator==` (const char \*, [Rcpp::Na\\_Proxy](#))
- bool `operator==` ([Rcpp::internal::string\\_proxy](#)< STRSXP > x, [Rcpp::Na\\_Proxy](#))
- bool `operator==` ([Rcpp::internal::const\\_string\\_proxy](#)< STRSXP > x, [Rcpp::Na\\_Proxy](#))
- bool `operator==` ([Rcpp::Na\\_Proxy](#), double x)
- bool `operator==` ([Rcpp::Na\\_Proxy](#), int x)
- bool `operator==` ([Rcpp::Na\\_Proxy](#), [Rcpp::String](#) x)
- bool `operator==` ([Rcpp::Na\\_Proxy](#), SEXP x)
- bool `operator==` ([Rcpp::Na\\_Proxy](#), [Rcomplex](#) x)
- bool `operator==` ([Rcpp::Na\\_Proxy](#), std::string)
- bool `operator==` ([Rcpp::Na\\_Proxy](#), const char \*)
- bool `operator==` ([Rcpp::Na\\_Proxy](#), [Rcpp::internal::string\\_proxy](#)< STRSXP > x)
- bool `operator==` ([Rcpp::Na\\_Proxy](#), [Rcpp::internal::const\\_string\\_proxy](#)< STRSXP > x)

### 6.424.1 Detailed Description

Definition at line 22 of file `Na_Proxy.h`.

### 6.424.2 Friends And Related Function Documentation

#### 6.424.2.1 `operator==` [1/18]

```
bool operator== (
    const char * ,
    Rcpp::Na\_Proxy ) [friend]
```

Definition at line 30 of file `Na_Proxy.h`.

#### 6.424.2.2 `operator==` [2/18]

```
bool operator== (
    double x,
    Rcpp::Na\_Proxy ) [friend]
```

Definition at line 24 of file `Na_Proxy.h`.

**6.424.2.3 operator==** [3/18]

```
bool operator== (
    int x,
    Rcpp::Na_Proxy ) [friend]
```

Definition at line 25 of file Na\_Proxy.h.

**6.424.2.4 operator==** [4/18]

```
bool operator== (
    Rcomplex x,
    Rcpp::Na_Proxy ) [friend]
```

Definition at line 27 of file Na\_Proxy.h.

**6.424.2.5 operator==** [5/18]

```
bool operator== (
    Rcpp::internal::const_string_proxy< STRSXP > x,
    Rcpp::Na_Proxy ) [friend]
```

Definition at line 34 of file Na\_Proxy.h.

**6.424.2.6 operator==** [6/18]

```
bool operator== (
    Rcpp::internal::string_proxy< STRSXP > x,
    Rcpp::Na_Proxy ) [friend]
```

Definition at line 31 of file Na\_Proxy.h.

**6.424.2.7 operator==** [7/18]

```
bool operator== (
    Rcpp::Na_Proxy ,
    const char * ) [friend]
```

Definition at line 44 of file Na\_Proxy.h.

**6.424.2.8 operator== [8/18]**

```
bool operator== (
    Rcpp::Na_Proxy ,
    double x ) [friend]
```

Definition at line 38 of file Na\_Proxy.h.

**6.424.2.9 operator== [9/18]**

```
bool operator== (
    Rcpp::Na_Proxy ,
    int x ) [friend]
```

Definition at line 39 of file Na\_Proxy.h.

**6.424.2.10 operator== [10/18]**

```
bool operator== (
    Rcpp::Na_Proxy ,
    Rcomplex x ) [friend]
```

Definition at line 42 of file Na\_Proxy.h.

**6.424.2.11 operator== [11/18]**

```
bool operator== (
    Rcpp::Na_Proxy ,
    Rcpp::internal::const_string_proxy< STRSXP > x ) [friend]
```

Definition at line 48 of file Na\_Proxy.h.

**6.424.2.12 operator== [12/18]**

```
bool operator== (
    Rcpp::Na_Proxy ,
    Rcpp::internal::string_proxy< STRSXP > x ) [friend]
```

Definition at line 45 of file Na\_Proxy.h.

**6.424.2.13 operator==** [13/18]

```
bool operator== (
    Rcpp::Na_Proxy ,
    Rcpp::String x ) [friend]
```

Definition at line 40 of file Na\_Proxy.h.

**6.424.2.14 operator==** [14/18]

```
bool operator== (
    Rcpp::Na_Proxy ,
    SEXP x ) [friend]
```

Definition at line 41 of file Na\_Proxy.h.

**6.424.2.15 operator==** [15/18]

```
bool operator== (
    Rcpp::Na_Proxy ,
    std::string ) [friend]
```

Definition at line 43 of file Na\_Proxy.h.

**6.424.2.16 operator==** [16/18]

```
bool operator== (
    Rcpp::String x,
    Rcpp::Na_Proxy ) [friend]
```

Definition at line 26 of file Na\_Proxy.h.

**6.424.2.17 operator==** [17/18]

```
bool operator== (
    SEXP x,
    Rcpp::Na_Proxy ) [friend]
```

Definition at line 28 of file Na\_Proxy.h.



### 6.424.2.18 operator== [18/18]

```
bool operator== (
    std::string ,
    Rcpp::Na_Proxy ) [friend]
```

Definition at line 29 of file Na\_Proxy.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Na\\_Proxy.h](#)

## 6.425 Rcpp::traits::named\_object< T > Class Template Reference

```
#include <named_object.h>
```

### Public Member Functions

- [named\\_object](#) (const std::string &name\_, const T &o\_)

### Public Attributes

- const std::string & [name](#)
- const T & [object](#)

### 6.425.1 Detailed Description

```
template<typename T>
class Rcpp::traits::named_object< T >
```

Definition at line 33 of file named\_object.h.

### 6.425.2 Constructor & Destructor Documentation

#### 6.425.2.1 named\_object()

```
template<typename T >
Rcpp::traits::named_object< T >::named_object (
    const std::string & name_,
    const T & o_ ) [inline]
```

Definition at line 35 of file named\_object.h.

## 6.425.3 Member Data Documentation

### 6.425.3.1 name

```
template<typename T >
const std::string& Rcpp::traits::named_object< T >::name
```

Definition at line 37 of file named\_object.h.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=().

### 6.425.3.2 object

```
template<typename T >
const T& Rcpp::traits::named_object< T >::object
```

Definition at line 38 of file named\_object.h.

Referenced by Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy::operator=().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/named\_object.h

## 6.426 Rcpp::traits::named\_object< SEXP > Class Reference

```
#include <named_object.h>
```

### Public Member Functions

- [named\\_object](#) (const std::string &name\_, const SEXP &o\_)
- [named\\_object](#) (const [named\\_object](#)< SEXP > &other)
- [~named\\_object](#) ()

### Public Attributes

- const std::string & [name](#)
- SEXP [object](#)

## Private Attributes

- SEXP [token](#)

### 6.426.1 Detailed Description

Definition at line 40 of file named\_object.h.

### 6.426.2 Constructor & Destructor Documentation

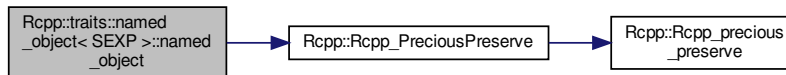
#### 6.426.2.1 named\_object() [1/2]

```
Rcpp::traits::named_object< SEXP >::named_object (
    const std::string & name_,
    const SEXP & o_ ) [inline]
```

Definition at line 42 of file named\_object.h.

References [Rcpp::Rcpp\\_PreciousPreserve\(\)](#).

Here is the call graph for this function:



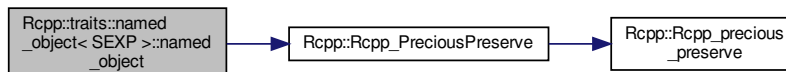
#### 6.426.2.2 named\_object() [2/2]

```
Rcpp::traits::named_object< SEXP >::named_object (
    const named_object< SEXP > & other ) [inline]
```

Definition at line 47 of file named\_object.h.

References [Rcpp::Rcpp\\_PreciousPreserve\(\)](#).

Here is the call graph for this function:



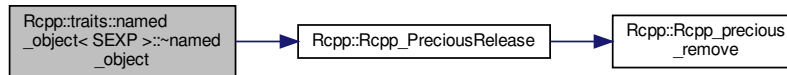
### 6.426.2.3 ~named\_object()

```
Rcpp::traits::named_object< SEXP >::~~named_object ( ) [inline]
```

Definition at line 51 of file named\_object.h.

References Rcpp::Rcpp\_PreciousRelease().

Here is the call graph for this function:



## 6.426.3 Member Data Documentation

### 6.426.3.1 name

```
const std::string& Rcpp::traits::named_object< SEXP >::name
```

Definition at line 55 of file named\_object.h.

### 6.426.3.2 object

```
SEXP Rcpp::traits::named_object< SEXP >::object
```

Definition at line 56 of file named\_object.h.

### 6.426.3.3 token

```
SEXP Rcpp::traits::named_object< SEXP >::token [private]
```

Definition at line 58 of file named\_object.h.

The documentation for this class was generated from the following file:

- `inst/include/Rcpp/traits/named_object.h`

## 6.427 Rcpp::internal::NamedPlaceholder Class Reference

```
#include <Named.h>
```

### Public Member Functions

- [NamedPlaceholder](#) ()
- [~NamedPlaceholder](#) ()
- [Argument operator\[\]](#) (const std::string &arg) const
- [Argument operator\(\)](#) (const std::string &arg) const
- [operator SEXP](#) () const

### 6.427.1 Detailed Description

Definition at line 50 of file Named.h.

### 6.427.2 Constructor & Destructor Documentation

#### 6.427.2.1 NamedPlaceholder()

```
Rcpp::internal::NamedPlaceholder::NamedPlaceholder ( ) [inline]
```

Definition at line 52 of file Named.h.

#### 6.427.2.2 ~NamedPlaceholder()

```
Rcpp::internal::NamedPlaceholder::~~NamedPlaceholder ( ) [inline]
```

Definition at line 53 of file Named.h.

### 6.427.3 Member Function Documentation

**6.427.3.1 operator SEXP()**

```
Rcpp::internal::NamedPlaceHolder::operator SEXP ( ) const [inline]
```

Definition at line 60 of file Named.h.

**6.427.3.2 operator>()()**

```
Argument Rcpp::internal::NamedPlaceHolder::operator() (
    const std::string & arg ) const [inline]
```

Definition at line 57 of file Named.h.

**6.427.3.3 operator[]()**

```
Argument Rcpp::internal::NamedPlaceHolder::operator[] (
    const std::string & arg ) const [inline]
```

Definition at line 54 of file Named.h.

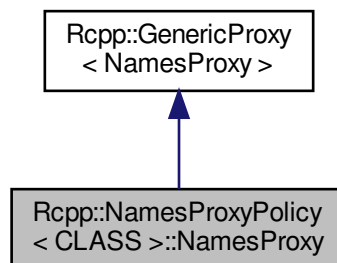
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Named.h](#)

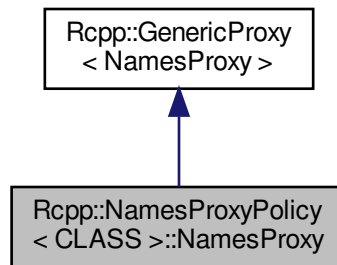
**6.428 Rcpp::NamesProxyPolicy< CLASS >::NamesProxy Class Reference**

```
#include <NamesProxy.h>
```

Inheritance diagram for Rcpp::NamesProxyPolicy< CLASS >::NamesProxy:



Collaboration diagram for Rcpp::NamesProxyPolicy< CLASS >::NamesProxy:



## Public Member Functions

- [NamesProxy](#) (CLASS &v)
- [NamesProxy](#) & [operator=](#) (const [NamesProxy](#) &rhs)
- template<typename T >  
[NamesProxy](#) & [operator=](#) (const T &rhs)
- template<typename T >  
[operator T](#) () const
- template<typename T >  
[NamesProxyPolicy](#)< CLASS >::[NamesProxy](#) & [operator=](#) (const T &rhs)

## Private Member Functions

- SEXP [get](#) () const
- void [set](#) (SEXP x)

## Private Attributes

- CLASS & [parent](#)

### 6.428.1 Detailed Description

```

template<typename CLASS>
class Rcpp::NamesProxyPolicy< CLASS >::NamesProxy

```

Definition at line 27 of file NamesProxy.h.

## 6.428.2 Constructor & Destructor Documentation

### 6.428.2.1 NamesProxy()

```
template<typename CLASS >  
Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::NamesProxy (  
    CLASS & v ) [inline]
```

Definition at line 29 of file NamesProxy.h.

## 6.428.3 Member Function Documentation

### 6.428.3.1 get()

```
template<typename CLASS >  
SEXPR Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::get ( ) const [inline], [private]
```

Definition at line 45 of file NamesProxy.h.

References `Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::parent`, and `RCPP_GET_NAMES`.

Referenced by `Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator=()`.

### 6.428.3.2 operator T()

```
template<typename CLASS >  
template<typename T >  
Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator T
```

Definition at line 70 of file proxy.h.



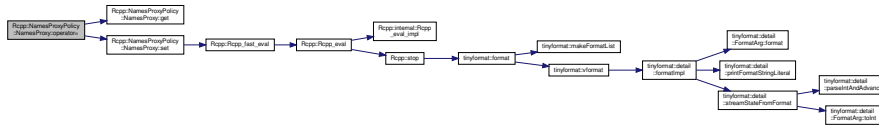
## 6.428.3.3 operator=() [1/3]

```
template<typename CLASS >
NamesProxy& Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator= (
    const NamesProxy & rhs ) [inline]
```

Definition at line 32 of file NamesProxy.h.

References Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::get(), and Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::set().

Here is the call graph for this function:



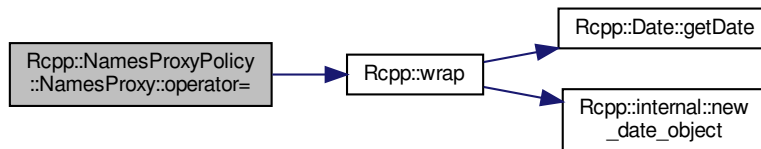
## 6.428.3.4 operator=() [2/3]

```
template<typename CLASS >
template<typename T >
NamesProxyPolicy<CLASS>::NamesProxy& Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator= (
    const T & rhs )
```

Definition at line 63 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:



### 6.428.3.5 operator=() [3/3]

```
template<typename CLASS >
template<typename T >
NamesProxy & Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator= (
    const T & rhs )
```

### 6.428.3.6 set()

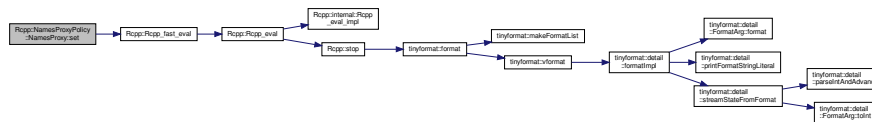
```
template<typename CLASS >
void Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::set (
    SEXP x ) [inline], [private]
```

Definition at line 49 of file NamesProxy.h.

References Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::parent, and Rcpp::Rcpp\_fast\_eval().

Referenced by Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::operator=().

Here is the call graph for this function:



## 6.428.4 Member Data Documentation

### 6.428.4.1 parent

```
template<typename CLASS >
CLASS & Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::parent [private]
```

Definition at line 43 of file NamesProxy.h.

Referenced by Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::get(), and Rcpp::NamesProxyPolicy< CLASS >::NamesProxy::set().

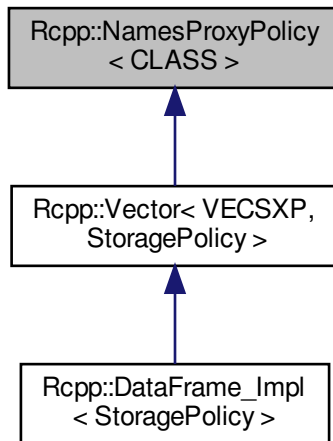
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/NamesProxy.h
- inst/include/Rcpp/api/meat/proxy.h

## 6.429 Rcpp::NamesProxyPolicy< CLASS > Class Template Reference

```
#include <NamesProxy.h>
```

Inheritance diagram for Rcpp::NamesProxyPolicy< CLASS >:



### Classes

- class [const\\_NamesProxy](#)
- class [NamesProxy](#)

### Public Member Functions

- [NamesProxy names \(\)](#)
- [const\\_NamesProxy names \(\) const](#)

### 6.429.1 Detailed Description

```
template<typename CLASS>  
class Rcpp::NamesProxyPolicy< CLASS >
```

Definition at line 24 of file NamesProxy.h.

## 6.429.2 Member Function Documentation

### 6.429.2.1 names() [1/2]

```
template<typename CLASS >
NamesProxy Rcpp::NamesProxyPolicy< CLASS >::names ( ) [inline]
```

Definition at line 82 of file NamesProxy.h.

Referenced by Rcpp::Module::classes\_info(), class\_< Class >::fields(), Rcpp::DataFrame\_Impl< StoragePolicy >::from\_list(), Rcpp::Module::functions\_arity(), class\_< Class >::getMethods(), lapplyCpp(), class\_< Class >::methods\_arity(), class\_< Class >::methods\_voidness(), and class\_< Class >::property\_classes().

### 6.429.2.2 names() [2/2]

```
template<typename CLASS >
const NamesProxy Rcpp::NamesProxyPolicy< CLASS >::names ( ) const [inline]
```

Definition at line 86 of file NamesProxy.h.

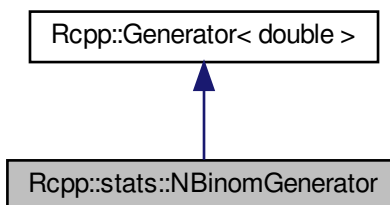
The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/NamesProxy.h

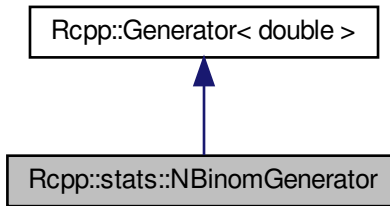
## 6.430 Rcpp::stats::NBinomGenerator Class Reference

```
#include <rnbinom.h>
```

Inheritance diagram for Rcpp::stats::NBinomGenerator:



Collaboration diagram for Rcpp::stats::NBinomGenerator:



## Public Member Functions

- [NBinomGenerator](#) (double *siz\_*, double *prob\_*)
- double [operator\(\)](#) () const

## Private Attributes

- double [siz](#)
- double [lambda](#)

## Additional Inherited Members

### 6.430.1 Detailed Description

Definition at line 28 of file `rnbinom.h`.

### 6.430.2 Constructor & Destructor Documentation

#### 6.430.2.1 NBinomGenerator()

```
Rcpp::stats::NBinomGenerator::NBinomGenerator (  
    double siz_,  
    double prob_ ) [inline]
```

Definition at line 31 of file `rnbinom.h`.

### 6.430.3 Member Function Documentation

#### 6.430.3.1 operator()

```
double Rcpp::stats::NBinomGenerator::operator() ( ) const [inline]
```

Definition at line 34 of file `rnbinom.h`.

References `lambda`, and `siz`.

### 6.430.4 Member Data Documentation

#### 6.430.4.1 lambda

```
double Rcpp::stats::NBinomGenerator::lambda [private]
```

Definition at line 40 of file `rnbinom.h`.

Referenced by `operator()`.

#### 6.430.4.2 siz

```
double Rcpp::stats::NBinomGenerator::siz [private]
```

Definition at line 39 of file `rnbinom.h`.

Referenced by `operator()`.

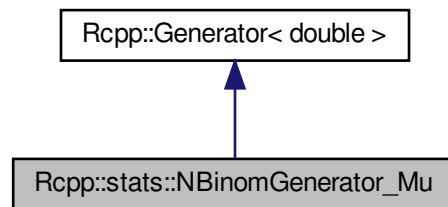
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rnbinom.h](#)

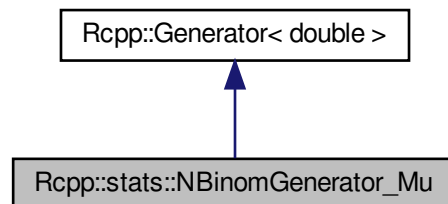
## 6.431 Rcpp::stats::NBinomGenerator\_Mu Class Reference

```
#include <rnbinom_mu.h>
```

Inheritance diagram for Rcpp::stats::NBinomGenerator\_Mu:



Collaboration diagram for Rcpp::stats::NBinomGenerator\_Mu:



### Public Member Functions

- [NBinomGenerator\\_Mu](#) (double siz\_, double mu\_)
- double [operator\(\)](#) () const

### Private Attributes

- double [siz](#)
- double [lambda](#)

## Additional Inherited Members

### 6.431.1 Detailed Description

Definition at line 28 of file `rnbinom_mu.h`.

### 6.431.2 Constructor & Destructor Documentation

#### 6.431.2.1 NBinomGenerator\_Mu()

```
Rcpp::stats::NBinomGenerator_Mu::NBinomGenerator_Mu (
    double siz_,
    double mu_ ) [inline]
```

Definition at line 31 of file `rnbinom_mu.h`.

### 6.431.3 Member Function Documentation

#### 6.431.3.1 operator>()()

```
double Rcpp::stats::NBinomGenerator_Mu::operator() ( ) const [inline]
```

Definition at line 34 of file `rnbinom_mu.h`.

References `lambda`, and `siz`.

### 6.431.4 Member Data Documentation

#### 6.431.4.1 lambda

```
double Rcpp::stats::NBinomGenerator_Mu::lambda [private]
```

Definition at line 40 of file `rnbinom_mu.h`.

Referenced by `operator>()()`.



### 6.431.4.2 siz

```
double Rcpp::stats::NBinomGenerator_Mu::siz [private]
```

Definition at line 39 of file `rnbinom_mu.h`.

Referenced by `operator()`.

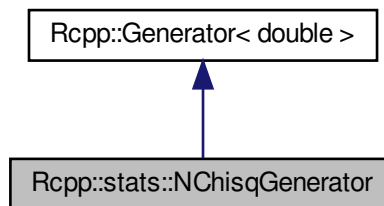
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rnbinom\\_mu.h](#)

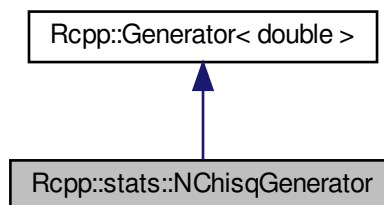
## 6.432 Rcpp::stats::NChisqGenerator Class Reference

```
#include <rnchisq.h>
```

Inheritance diagram for `Rcpp::stats::NChisqGenerator`:



Collaboration diagram for `Rcpp::stats::NChisqGenerator`:



## Public Member Functions

- [NChisqGenerator](#) (double *df\_*, double *lambda\_*)
- double [operator\(\)](#) () const

## Private Attributes

- double [df](#)
- double [df\\_2](#)
- double [lambda\\_2](#)

## Additional Inherited Members

### 6.432.1 Detailed Description

Definition at line 28 of file `rnchisq.h`.

### 6.432.2 Constructor & Destructor Documentation

#### 6.432.2.1 NChisqGenerator()

```
Rcpp::stats::NChisqGenerator::NChisqGenerator (  
    double df_,  
    double lambda_ ) [inline]
```

Definition at line 31 of file `rnchisq.h`.

### 6.432.3 Member Function Documentation

#### 6.432.3.1 operator()()

```
double Rcpp::stats::NChisqGenerator::operator() ( ) const [inline]
```

Definition at line 34 of file `rnchisq.h`.

References [df](#), [df\\_2](#), and [lambda\\_2](#).

## 6.432.4 Member Data Documentation

### 6.432.4.1 df

```
double Rcpp::stats::NChisqGenerator::df [private]
```

Definition at line 46 of file `rnchisq.h`.

Referenced by `operator()`.

### 6.432.4.2 df\_2

```
double Rcpp::stats::NChisqGenerator::df_2 [private]
```

Definition at line 47 of file `rnchisq.h`.

Referenced by `operator()`.

### 6.432.4.3 lambda\_2

```
double Rcpp::stats::NChisqGenerator::lambda_2 [private]
```

Definition at line 48 of file `rnchisq.h`.

Referenced by `operator()`.

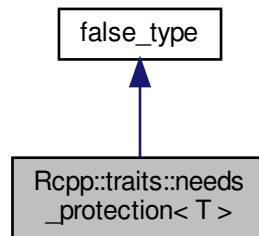
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rnchisq.h](#)

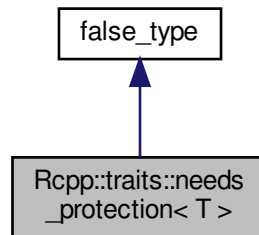
## 6.433 Rcpp::traits::needs\_protection< T > Struct Template Reference

```
#include <named_object.h>
```

Inheritance diagram for Rcpp::traits::needs\_protection< T >:



Collaboration diagram for Rcpp::traits::needs\_protection< T >:



### Additional Inherited Members

#### 6.433.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::needs_protection< T >
```

Definition at line 30 of file named\_object.h.

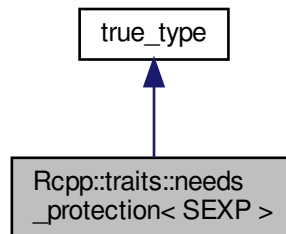
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/named\\_object.h](#)

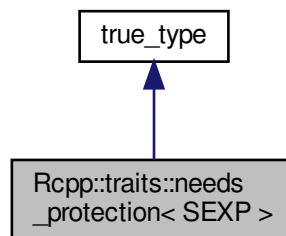
## 6.434 Rcpp::traits::needs\_protection< SEXP > Struct Reference

```
#include <named_object.h>
```

Inheritance diagram for Rcpp::traits::needs\_protection< SEXP >:



Collaboration diagram for Rcpp::traits::needs\_protection< SEXP >:



### Additional Inherited Members

#### 6.434.1 Detailed Description

Definition at line 31 of file named\_object.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/named\_object.h

## 6.435 Rcpp::sugar::negate< NA > Struct Template Reference

```
#include <not.h>
```

### Static Public Member Functions

- static int [apply](#) (int x)

#### 6.435.1 Detailed Description

```
template<bool NA>
struct Rcpp::sugar::negate< NA >
```

Definition at line 28 of file not.h.

#### 6.435.2 Member Function Documentation

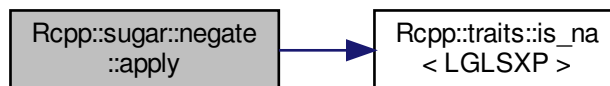
##### 6.435.2.1 apply()

```
template<bool NA>
static int Rcpp::sugar::negate< NA >::apply (
    int x ) [inline], [static]
```

Definition at line 29 of file not.h.

References [Rcpp::traits::is\\_na< LGLSXP >\(\)](#).

Here is the call graph for this function:



The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/logical/not.h](#)

## 6.436 Rcpp::sugar::negate< false > Struct Reference

```
#include <not.h>
```

### Static Public Member Functions

- static int [apply](#) (int x)

### 6.436.1 Detailed Description

Definition at line 34 of file not.h.

### 6.436.2 Member Function Documentation

#### 6.436.2.1 [apply\(\)](#)

```
static int Rcpp::sugar::negate< false >::apply (  
    int x ) [inline], [static]
```

Definition at line 35 of file not.h.

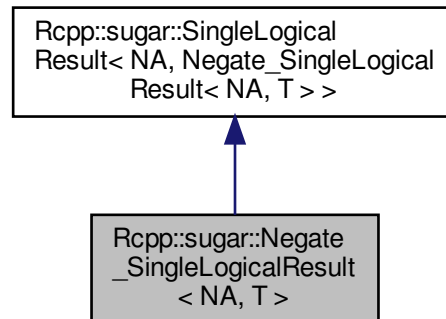
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/logical/[not.h](#)

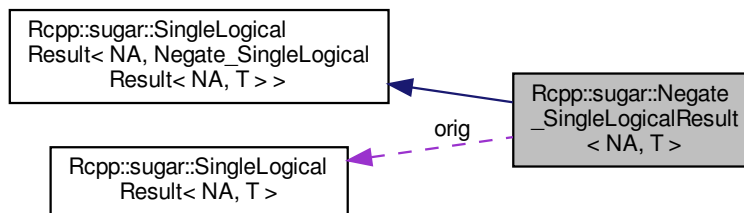
## 6.437 Rcpp::sugar::Negate\_SingleLogicalResult< NA, T > Class Template Reference

```
#include <not.h>
```

Inheritance diagram for `Rcpp::sugar::Negate_SingleLogicalResult< NA, T >`:



Collaboration diagram for `Rcpp::sugar::Negate_SingleLogicalResult< NA, T >`:



## Public Types

- typedef `SingleLogicalResult< NA, T > TYPE`
- typedef `SingleLogicalResult< NA, Negate_SingleLogicalResult< NA, T > > BASE`

## Public Member Functions

- `Negate_SingleLogicalResult` (const `TYPE` &orig\_)
- void `apply` ()

## Private Attributes

- const `TYPE` & `orig`



## Additional Inherited Members

### 6.437.1 Detailed Description

```
template<bool NA, typename T>
class Rcpp::sugar::Negate_SingleLogicalResult< NA, T >
```

Definition at line 42 of file not.h.

### 6.437.2 Member Typedef Documentation

#### 6.437.2.1 BASE

```
template<bool NA, typename T >
typedef SingleLogicalResult<NA, Negate_SingleLogicalResult<NA,T> > Rcpp::sugar::Negate_SingleLogicalResult<
NA, T >::BASE
```

Definition at line 45 of file not.h.

#### 6.437.2.2 TYPE

```
template<bool NA, typename T >
typedef SingleLogicalResult<NA,T> Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::TYPE
```

Definition at line 44 of file not.h.

### 6.437.3 Constructor & Destructor Documentation

#### 6.437.3.1 Negate\_SingleLogicalResult()

```
template<bool NA, typename T >
Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::Negate_SingleLogicalResult (
    const TYPE & orig_ ) [inline]
```

Definition at line 46 of file not.h.

## 6.437.4 Member Function Documentation

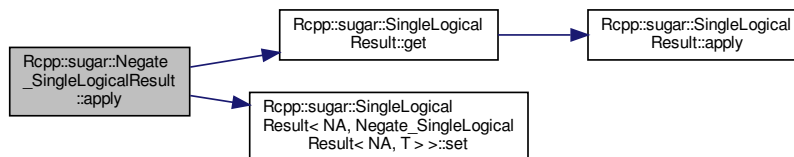
### 6.437.4.1 apply()

```
template<bool NA, typename T >
void Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::apply ( ) [inline]
```

Definition at line 48 of file not.h.

References `Rcpp::sugar::SingleLogicalResult< NA, T >::get()`, `Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::orig`, and `Rcpp::sugar::SingleLogicalResult< NA, Negate_SingleLogicalResult< NA, T > >::set()`.

Here is the call graph for this function:



## 6.437.5 Member Data Documentation

### 6.437.5.1 orig

```
template<bool NA, typename T >
const TYPE& Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::orig [private]
```

Definition at line 53 of file not.h.

Referenced by `Rcpp::sugar::Negate_SingleLogicalResult< NA, T >::apply()`.

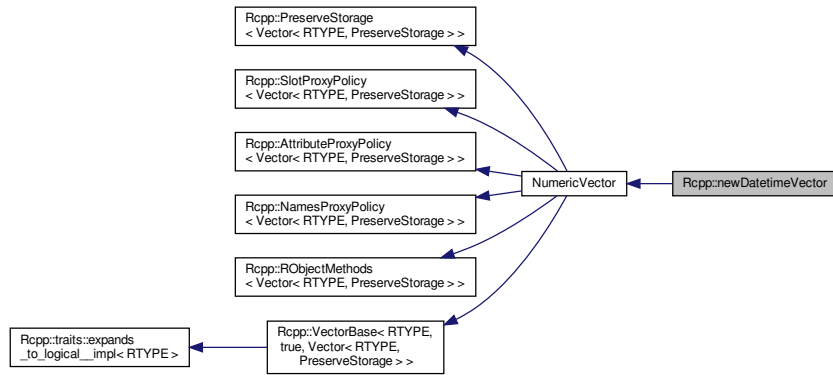
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/logical/not.h](#)

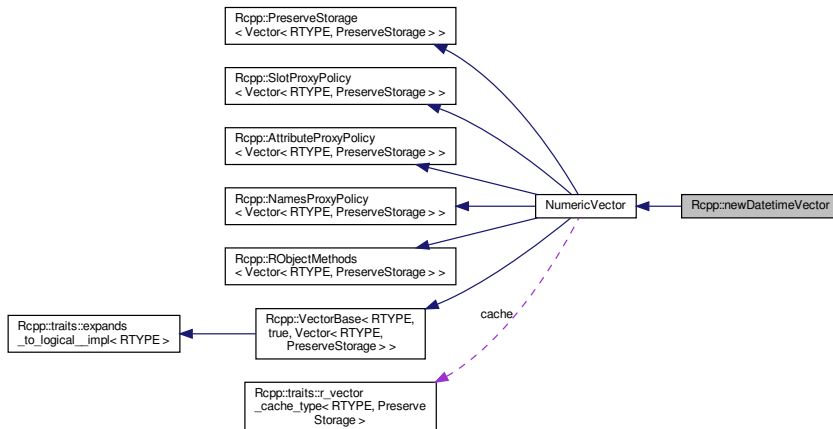
## 6.438 Rcpp::newDatetimeVector Class Reference

```
#include <newDatetimeVector.h>
```

Inheritance diagram for Rcpp::newDatetimeVector:



Collaboration diagram for Rcpp::newDatetimeVector:



### Public Member Functions

- `template<int RTYPE, bool NA, typename VEC >`  
[newDatetimeVector](#) (const [VectorBase](#)< RTYPE, NA, VEC > &other, const char \*tz="")
- [newDatetimeVector](#) (SEXP vec, const char \*tz="")
- [newDatetimeVector](#) (int n, const char \*tz="")
- `std::vector< Datetime >` [getDatetimes](#) () const
- [newDatetimeVector](#) & `operator=` (const [newDatetimeVector](#) &rhs)

## Private Member Functions

- void [setClass](#) (const char \*tz)

## Friends

- std::ostream & [operator<<](#) (std::ostream &s, const [newDatetimeVector](#) d)

## Additional Inherited Members

### 6.438.1 Detailed Description

Definition at line 29 of file newDatetimeVector.h.

### 6.438.2 Constructor & Destructor Documentation

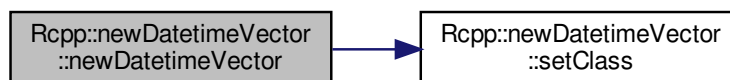
#### 6.438.2.1 newDatetimeVector() [1/3]

```
template<int RTYPE, bool NA, typename VEC >  
Rcpp::newDatetimeVector::newDatetimeVector (  
    const VectorBase< RTYPE, NA, VEC > & other,  
    const char * tz = "" ) [inline]
```

Definition at line 32 of file newDatetimeVector.h.

References [setClass\(\)](#).

Here is the call graph for this function:



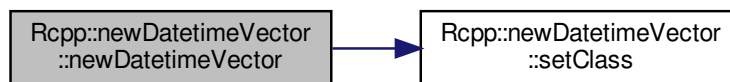
### 6.438.2.2 newDatetimeVector() [2/3]

```
Rcpp::newDatetimeVector::newDatetimeVector (
    SEXP vec,
    const char * tz = "" ) [inline]
```

Definition at line 37 of file newDatetimeVector.h.

References `setClass()`.

Here is the call graph for this function:



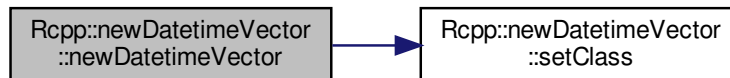
### 6.438.2.3 newDatetimeVector() [3/3]

```
Rcpp::newDatetimeVector::newDatetimeVector (
    int n,
    const char * tz = "" ) [inline]
```

Definition at line 42 of file newDatetimeVector.h.

References `setClass()`.

Here is the call graph for this function:



## 6.438.3 Member Function Documentation

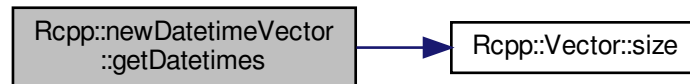
### 6.438.3.1 getDatetimes()

```
std::vector<Datetime> Rcpp::newDatetimeVector::getDatetimes ( ) const [inline]
```

Definition at line 47 of file newDatetimeVector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



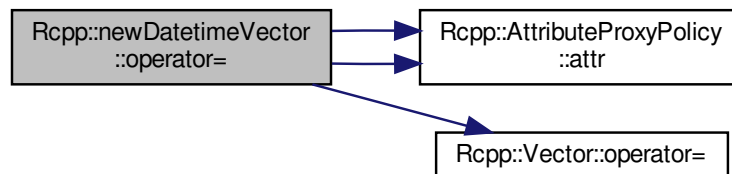
### 6.438.3.2 operator=()

```
newDatetimeVector& Rcpp::newDatetimeVector::operator= (
    const newDatetimeVector & rhs ) [inline]
```

Definition at line 55 of file newDatetimeVector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::AttributeProxyPolicy< Vector< RTYPE, PreserveStorage > >::attr(), and Rcpp::Vector< RTYPE, StoragePolicy >::operator=().

Here is the call graph for this function:



### 6.438.3.3 setClass()

```
void Rcpp::newDatetimeVector::setClass (
    const char * tz ) [inline], [private]
```

Definition at line 68 of file newDatetimeVector.h.

Referenced by newDatetimeVector().

## 6.438.4 Friends And Related Function Documentation

### 6.438.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & s,
    const newDatetimeVector d ) [friend]
```

Definition at line 81 of file newDatetimeVector.h.

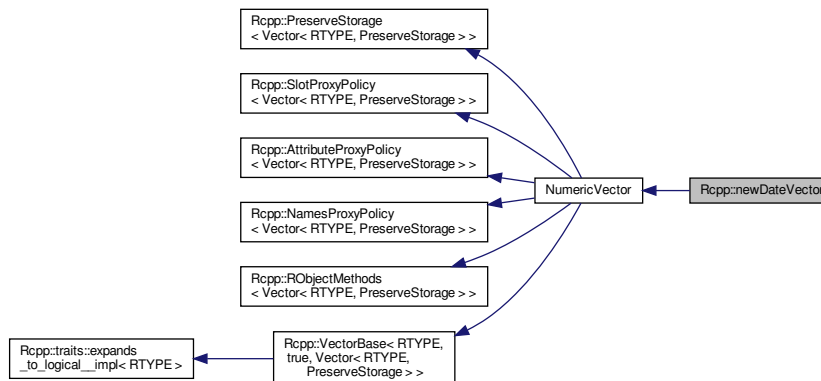
The documentation for this class was generated from the following file:

- inst/include/Rcpp/date\_datetime/newDatetimeVector.h

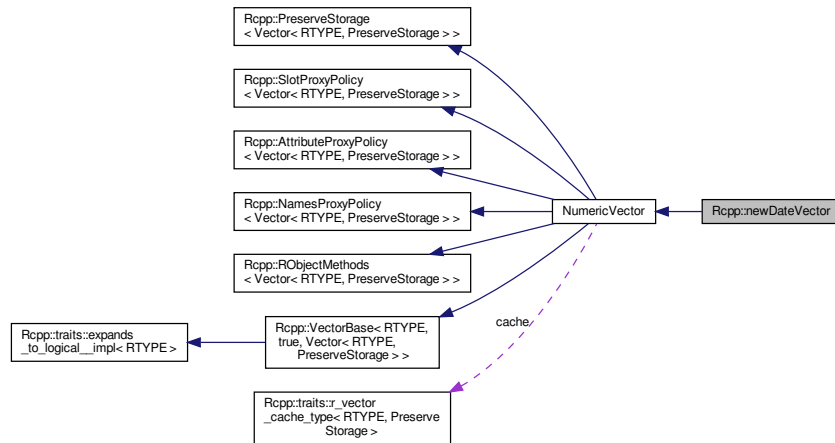
## 6.439 Rcpp::newDateVector Class Reference

```
#include <newDateVector.h>
```

Inheritance diagram for Rcpp::newDateVector:



Collaboration diagram for `Rcpp::newDateVector`:



## Public Member Functions

- `template<int RTYPE, bool NA, typename VEC >`  
`newDateVector` (const `VectorBase< RTYPE, NA, VEC >` &vec)
- `newDateVector` (SEXP vec)
- `newDateVector` (int n)
- `std::vector< Date >` `getDates` () const
- `newDateVector` & `operator=` (const `newDateVector` &rhs)

## Private Member Functions

- void `setClass` ()

## Friends

- `std::ostream` & `operator<<` (`std::ostream` &s, const `newDateVector` d)

## Additional Inherited Members

### 6.439.1 Detailed Description

Definition at line 29 of file `newDateVector.h`.

### 6.439.2 Constructor & Destructor Documentation



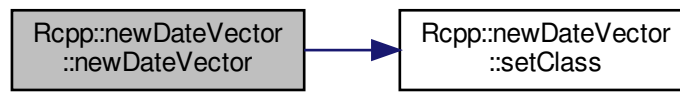
### 6.439.2.1 newDateVector() [1/3]

```
template<int RTYPE, bool NA, typename VEC >  
Rcpp::newDateVector::newDateVector (  
    const VectorBase< RTYPE, NA, VEC > & vec ) [inline]
```

Definition at line 32 of file newDateVector.h.

References `setClass()`.

Here is the call graph for this function:



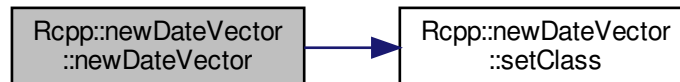
### 6.439.2.2 newDateVector() [2/3]

```
Rcpp::newDateVector::newDateVector (  
    SEXP vec ) [inline]
```

Definition at line 36 of file newDateVector.h.

References `setClass()`.

Here is the call graph for this function:



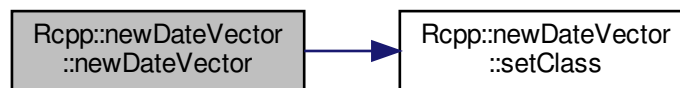
### 6.439.2.3 newDateVector() [3/3]

```
Rcpp::newDateVector::newDateVector (
    int n ) [inline]
```

Definition at line 37 of file newDateVector.h.

References setClass().

Here is the call graph for this function:



## 6.439.3 Member Function Documentation

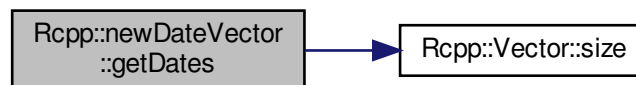
### 6.439.3.1 getDates()

```
std::vector<Date> Rcpp::newDateVector::getDates ( ) const [inline]
```

Definition at line 39 of file newDateVector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



### 6.439.3.2 operator=()

```
newDateVector& Rcpp::newDateVector::operator= (
    const newDateVector & rhs ) [inline]
```

Definition at line 47 of file newDateVector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::operator=().

Here is the call graph for this function:



### 6.439.3.3 setClass()

```
void Rcpp::newDateVector::setClass ( ) [inline], [private]
```

Definition at line 59 of file newDateVector.h.

Referenced by `newDateVector()`.

## 6.439.4 Friends And Related Function Documentation

### 6.439.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & s,
    const newDateVector d ) [friend]
```

Definition at line 65 of file newDateVector.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/date\\_datetime/newDateVector.h](#)

## 6.440 Rcpp::sugar::cbind\_impl::detail::has\_stored\_type< T >::no Struct Reference

### Public Attributes

- char [array](#) [2]

### 6.440.1 Detailed Description

```
template<typename T>
struct Rcpp::sugar::cbind_impl::detail::has_stored_type< T >::no
```

Definition at line 376 of file cbind.h.

### 6.440.2 Member Data Documentation

#### 6.440.2.1 array

```
template<typename T >
char Rcpp::sugar::cbind\_impl::detail::has\_stored\_type< T >::no::array[2]
```

Definition at line 377 of file cbind.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/cbind.h](#)

## 6.441 Rcpp::no\_init\_matrix Class Reference

```
#include <no_init.h>
```

### Public Member Functions

- [no\\_init\\_matrix](#) (int nr\_, int nc\_)
- int [nrow](#) () const
- int [ncol](#) () const
- [template<int RTYPE, template< class > class StoragePolicy> operator Matrix< RTYPE, StoragePolicy > \(\) const](#)

## Private Attributes

- int `nr`
- int `nc`

### 6.441.1 Detailed Description

Definition at line 51 of file `no_init.h`.

### 6.441.2 Constructor & Destructor Documentation

#### 6.441.2.1 `no_init_matrix()`

```
Rcpp::no_init_matrix::no_init_matrix (  
    int nr_,  
    int nc_ ) [inline]
```

Definition at line 53 of file `no_init.h`.

### 6.441.3 Member Function Documentation

#### 6.441.3.1 `ncol()`

```
int Rcpp::no_init_matrix::ncol ( ) const [inline]
```

Definition at line 59 of file `no_init.h`.

References `nc`.

#### 6.441.3.2 `nrow()`

```
int Rcpp::no_init_matrix::nrow ( ) const [inline]
```

Definition at line 55 of file `no_init.h`.

References `nr`.

### 6.441.3.3 operator Matrix< RTYPE, StoragePolicy >()

```
template<int RTYPE, template< class > class StoragePolicy>  
Rcpp::no_init_matrix::operator Matrix< RTYPE, StoragePolicy > ( ) const [inline]
```

Definition at line 64 of file no\_init.h.

References nc, and nr.

## 6.441.4 Member Data Documentation

### 6.441.4.1 nc

```
int Rcpp::no_init_matrix::nc [private]
```

Definition at line 74 of file no\_init.h.

Referenced by ncol(), and operator Matrix< RTYPE, StoragePolicy >().

### 6.441.4.2 nr

```
int Rcpp::no_init_matrix::nr [private]
```

Definition at line 73 of file no\_init.h.

Referenced by nrow(), and operator Matrix< RTYPE, StoragePolicy >().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/[no\\_init.h](#)

## 6.442 Rcpp::no\_init\_vector Class Reference

```
#include <no_init.h>
```

### Public Member Functions

- [no\\_init\\_vector](#) (R\_xlen\_t size\_)
- [R\\_xlen\\_t get](#) () const
- [template<int RTYPE, template< class > class StoragePolicy>  
operator Vector< RTYPE, StoragePolicy > \(\) const](#)

## Private Attributes

- [R\\_xlen\\_t size](#)

### 6.442.1 Detailed Description

Definition at line 30 of file no\_init.h.

### 6.442.2 Constructor & Destructor Documentation

#### 6.442.2.1 no\_init\_vector()

```
Rcpp::no_init_vector::no_init_vector (  
    R_xlen_t size_ ) [inline]
```

Definition at line 32 of file no\_init.h.

### 6.442.3 Member Function Documentation

#### 6.442.3.1 get()

```
R_xlen_t Rcpp::no_init_vector::get ( ) const [inline]
```

Definition at line 34 of file no\_init.h.

References [size](#).

Referenced by [Rcpp::Vector< RTYPE, StoragePolicy >::Vector\(\)](#).

#### 6.442.3.2 operator Vector< RTYPE, StoragePolicy >()

```
template<int RTYPE, template< class > class StoragePolicy>  
Rcpp::no_init_vector::operator Vector< RTYPE, StoragePolicy > ( ) const [inline]
```

Definition at line 39 of file no\_init.h.

References [size](#).

## 6.442.4 Member Data Documentation

### 6.442.4.1 size

```
R_xlen_t Rcpp::no_init_vector::size [private]
```

Definition at line 48 of file no\_init.h.

Referenced by `get()`, and operator `Vector< RTYPE, StoragePolicy >()`.

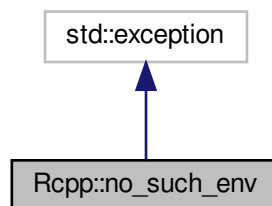
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/vector/no\\_init.h](#)

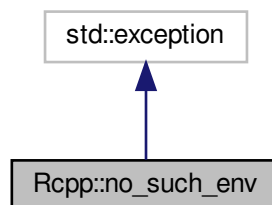
## 6.443 Rcpp::no\_such\_env Class Reference

```
#include <exceptions.h>
```

Inheritance diagram for `Rcpp::no_such_env`:



Collaboration diagram for `Rcpp::no_such_env`:





## Public Member Functions

- [no\\_such\\_env](#) (const std::string &name) throw ()
- [no\\_such\\_env](#) (int pos) throw ()
- virtual [~no\\_such\\_env](#) () throw ()
- virtual const char \* [what](#) () const throw ()

## Private Attributes

- std::string [message](#)

### 6.443.1 Detailed Description

Definition at line 70 of file exceptions.h.

### 6.443.2 Constructor & Destructor Documentation

#### 6.443.2.1 no\_such\_env() [1/2]

```
Rcpp::no_such_env::no_such_env (  
    const std::string & name ) throw ( )    [inline]
```

Definition at line 72 of file exceptions.h.

#### 6.443.2.2 no\_such\_env() [2/2]

```
Rcpp::no_such_env::no_such_env (  
    int pos ) throw ( )    [inline]
```

Definition at line 74 of file exceptions.h.

#### 6.443.2.3 ~no\_such\_env()

```
virtual Rcpp::no_such_env::~no_such_env ( ) throw ( )    [inline], [virtual]
```

Definition at line 76 of file exceptions.h.

### 6.443.3 Member Function Documentation

#### 6.443.3.1 what()

```
virtual const char* Rcpp::no_such_env::what ( ) const throw ( ) [inline], [virtual]
```

Definition at line 77 of file exceptions.h.

References message.

### 6.443.4 Member Data Documentation

#### 6.443.4.1 message

```
std::string Rcpp::no_such_env::message [private]
```

Definition at line 79 of file exceptions.h.

Referenced by what().

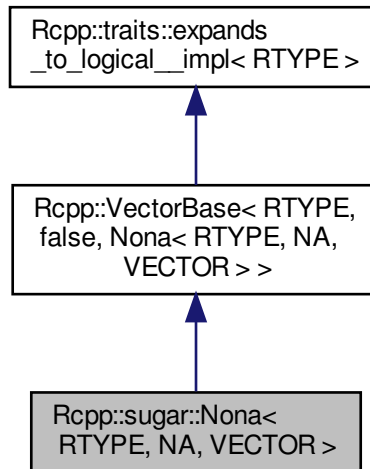
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/exceptions.h](#)

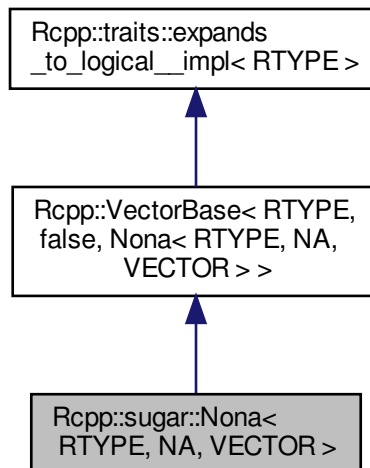
## 6.444 Rcpp::sugar::Nona< RTYPE, NA, VECTOR > Class Template Reference

```
#include <nona.h>
```

Inheritance diagram for Rcpp::sugar::Nona< RTYPE, NA, VECTOR >:



Collaboration diagram for Rcpp::sugar::Nona< RTYPE, NA, VECTOR >:



## Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, VECTOR >` `SUGAR_TYPE`

- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)

## Public Member Functions

- [Nona](#) (const [SUGAR\\_TYPE](#) &expr)
- [R\\_xlen\\_t size](#) () const
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const

## Private Attributes

- const VECTOR & [data](#)

### 6.444.1 Detailed Description

```
template<int RTYPE, bool NA, typename VECTOR>  
class Rcpp::sugar::Nona< RTYPE, NA, VECTOR >
```

Definition at line 29 of file nona.h.

### 6.444.2 Member Typedef Documentation

#### 6.444.2.1 STORAGE

```
template<int RTYPE, bool NA, typename VECTOR >  
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::STORAGE
```

Definition at line 32 of file nona.h.

#### 6.444.2.2 SUGAR\_TYPE

```
template<int RTYPE, bool NA, typename VECTOR >  
typedef Rcpp::VectorBase<RTYPE,NA,VECTOR> Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::SUGAR_TYPE
```

Definition at line 31 of file nona.h.

### 6.444.3 Constructor & Destructor Documentation

### 6.444.3.1 Nona()

```
template<int RTYPE, bool NA, typename VECTOR >
Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::Nona (
    const SUGAR_TYPE & expr ) [inline]
```

Definition at line 34 of file nona.h.

## 6.444.4 Member Function Documentation

### 6.444.4.1 operator[]()

```
template<int RTYPE, bool NA, typename VECTOR >
STORAGE Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 37 of file nona.h.

References Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::data.

### 6.444.4.2 size()

```
template<int RTYPE, bool NA, typename VECTOR >
R_xlen_t Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::size ( ) const [inline]
```

Definition at line 36 of file nona.h.

References Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::data.

## 6.444.5 Member Data Documentation

### 6.444.5.1 data

```
template<int RTYPE, bool NA, typename VECTOR >
const VECTOR& Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::data [private]
```

Definition at line 40 of file nona.h.

Referenced by Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::operator[](), Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >::operator[](), and Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::size().

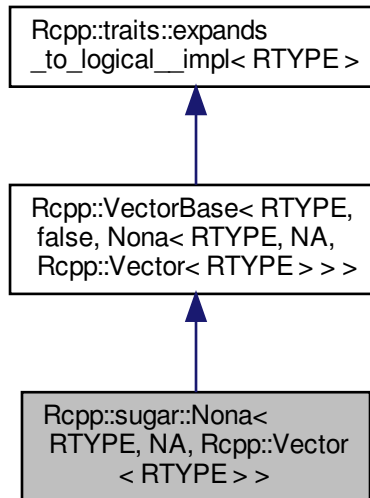
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/nona/nona.h](#)

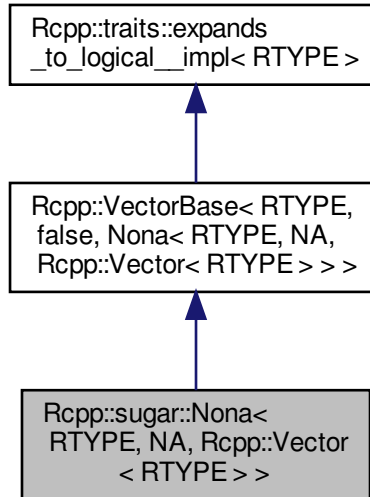
## 6.445 Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > > Class Template Reference

```
#include <nona.h>
```

Inheritance diagram for Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >:



Collaboration diagram for Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >:



## Public Types

- typedef [Rcpp::VectorBase< RTYPE, NA, Rcpp::Vector< RTYPE > >](#) [SUGAR\\_TYPE](#)
- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) [STORAGE](#)
- typedef [Rcpp::Vector< RTYPE >::const\\_iterator](#) [iterator](#)

## Public Member Functions

- [Nona](#) (const [SUGAR\\_TYPE](#) &expr)
- [R\\_xlen\\_t size](#) () const
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const

## Private Attributes

- [iterator data](#)
- [R\\_xlen\\_t n](#)

### 6.445.1 Detailed Description

```

template<int RTYPE, bool NA>
class Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >
  
```

Definition at line 45 of file nona.h.

## 6.445.2 Member Typedef Documentation

### 6.445.2.1 iterator

```
template<int RTYPE, bool NA>
typedef Rcpp::Vector<RTYPE>::const_iterator Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE >
>::iterator
```

Definition at line 49 of file nona.h.

### 6.445.2.2 STORAGE

```
template<int RTYPE, bool NA>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector<
RTYPE > >::STORAGE
```

Definition at line 48 of file nona.h.

### 6.445.2.3 SUGAR\_TYPE

```
template<int RTYPE, bool NA>
typedef Rcpp::VectorBase<RTYPE,NA, Rcpp::Vector<RTYPE> > Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector<
RTYPE > >::SUGAR_TYPE
```

Definition at line 47 of file nona.h.

## 6.445.3 Constructor & Destructor Documentation

### 6.445.3.1 Nona()

```
template<int RTYPE, bool NA>
Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >::Nona (
    const SUGAR_TYPE & expr ) [inline]
```

Definition at line 51 of file nona.h.



## 6.445.4 Member Function Documentation

### 6.445.4.1 operator[]()

```
template<int RTYPE, bool NA>  
STORAGE Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >::operator[] (   
    R_xlen_t i ) const [inline]
```

Definition at line 54 of file nona.h.

References Rcpp::sugar::Nona< RTYPE, NA, VECTOR >::data.

### 6.445.4.2 size()

```
template<int RTYPE, bool NA>  
R_xlen_t Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >::size ( ) const [inline]
```

Definition at line 53 of file nona.h.

## 6.445.5 Member Data Documentation

### 6.445.5.1 data

```
template<int RTYPE, bool NA>  
iterator Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >::data [private]
```

Definition at line 57 of file nona.h.

### 6.445.5.2 n

```
template<int RTYPE, bool NA>  
R_xlen_t Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >::n [private]
```

Definition at line 58 of file nona.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/nona/nona.h

## 6.446 Rcpp::sugar::NonaPrimitive< T > Class Template Reference

```
#include <nona.h>
```

### Public Member Functions

- [NonaPrimitive](#) (T t)
- [operator T](#) ()

### Private Attributes

- [T x](#)

### 6.446.1 Detailed Description

```
template<typename T>  
class Rcpp::sugar::NonaPrimitive< T >
```

Definition at line 62 of file nona.h.

### 6.446.2 Constructor & Destructor Documentation

#### 6.446.2.1 NonaPrimitive()

```
template<typename T >  
Rcpp::sugar::NonaPrimitive< T >::NonaPrimitive (  
    T t ) [inline]
```

Definition at line 64 of file nona.h.

### 6.446.3 Member Function Documentation

#### 6.446.3.1 operator T()

```
template<typename T >  
Rcpp::sugar::NonaPrimitive< T >::operator T ( ) [inline]
```

Definition at line 65 of file nona.h.

References [Rcpp::sugar::NonaPrimitive< T >::x](#).

## 6.446.4 Member Data Documentation

### 6.446.4.1 x

```
template<typename T >
T Rcpp::sugar::NonaPrimitive< T >::x [private]
```

Definition at line 68 of file nona.h.

Referenced by Rcpp::sugar::NonaPrimitive< T >::operator T().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/nona/nona.h](#)

## 6.447 Rcpp::NoProtectStorage< CLASS > Class Template Reference

```
#include <NoProtectStorage.h>
```

### Public Member Functions

- [NoProtectStorage](#) ()
- [~NoProtectStorage](#) ()
- void [set\\_\\_](#) (SEXP x)
- SEXP [get\\_\\_](#) () const
- SEXP [invalidate\\_\\_](#) ()
- CLASS & [copy\\_\\_](#) (const CLASS &other)
- bool [inherits](#) (const char \*clazz) const
- [operator SEXP](#) () const

### Private Attributes

- SEXP [data](#)

### 6.447.1 Detailed Description

```
template<typename CLASS>
class Rcpp::NoProtectStorage< CLASS >
```

Definition at line 7 of file NoProtectStorage.h.

## 6.447.2 Constructor & Destructor Documentation

### 6.447.2.1 NoProtectStorage()

```
template<typename CLASS >  
Rcpp::NoProtectStorage< CLASS >::NoProtectStorage ( ) [inline]
```

Definition at line 10 of file NoProtectStorage.h.

### 6.447.2.2 ~NoProtectStorage()

```
template<typename CLASS >  
Rcpp::NoProtectStorage< CLASS >::~~NoProtectStorage ( ) [inline]
```

Definition at line 12 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

## 6.447.3 Member Function Documentation

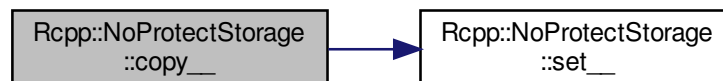
### 6.447.3.1 copy\_\_()

```
template<typename CLASS >  
CLASS& Rcpp::NoProtectStorage< CLASS >::copy__ (   
    const CLASS & other ) [inline]
```

Definition at line 33 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::set\_\_().

Here is the call graph for this function:



### 6.447.3.2 get\_\_()

```
template<typename CLASS >  
SEXP Rcpp::NoProtectStorage< CLASS >::get__ ( ) const [inline]
```

Definition at line 24 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

### 6.447.3.3 inherits()

```
template<typename CLASS >  
bool Rcpp::NoProtectStorage< CLASS >::inherits (  
    const char * clazz ) const [inline]
```

Definition at line 40 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

### 6.447.3.4 invalidate\_\_()

```
template<typename CLASS >  
SEXP Rcpp::NoProtectStorage< CLASS >::invalidate__ ( ) [inline]
```

Definition at line 28 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

### 6.447.3.5 operator SEXP()

```
template<typename CLASS >  
Rcpp::NoProtectStorage< CLASS >::operator SEXP ( ) const [inline]
```

Definition at line 44 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

### 6.447.3.6 set\_\_()

```
template<typename CLASS >
void Rcpp::NoProtectStorage< CLASS >::set__ (
    SEXP x ) [inline]
```

Definition at line 16 of file NoProtectStorage.h.

References Rcpp::NoProtectStorage< CLASS >::data.

Referenced by Rcpp::NoProtectStorage< CLASS >::copy\_\_().

## 6.447.4 Member Data Documentation

### 6.447.4.1 data

```
template<typename CLASS >
SEXP Rcpp::NoProtectStorage< CLASS >::data [private]
```

Definition at line 49 of file NoProtectStorage.h.

Referenced by Rcpp::NoProtectStorage< CLASS >::get\_\_(), Rcpp::NoProtectStorage< CLASS >::inherits(), Rcpp::NoProtectStorage< CLASS >::invalidate\_\_(), Rcpp::NoProtectStorage< CLASS >::operator SEXP(), Rcpp::NoProtectStorage< CLASS >::set\_\_(), and Rcpp::NoProtectStorage< CLASS >::~NoProtectStorage().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/storage/NoProtectStorage.h](#)

## 6.448 Rcpp::traits::normal\_wrap\_tag Struct Reference

```
#include <module_wrap_traits.h>
```

### 6.448.1 Detailed Description

Definition at line 29 of file module\_wrap\_traits.h.

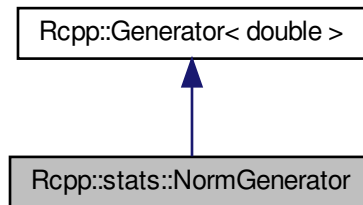
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/module\\_wrap\\_traits.h](#)

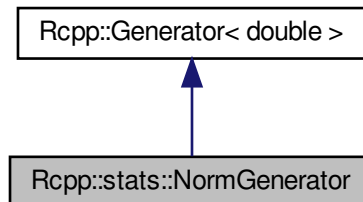
## 6.449 Rcpp::stats::NormGenerator Class Reference

```
#include <rnorm.h>
```

Inheritance diagram for Rcpp::stats::NormGenerator:



Collaboration diagram for Rcpp::stats::NormGenerator:



### Public Member Functions

- [NormGenerator](#) (double mean\_<sub>=</sub>0.0, double sd\_<sub>=</sub>1.0)
- double [operator\(\)](#) () const

### Private Attributes

- double [mean](#)
- double [sd](#)

## Additional Inherited Members

### 6.449.1 Detailed Description

Definition at line 28 of file rnorm.h.

### 6.449.2 Constructor & Destructor Documentation

#### 6.449.2.1 NormGenerator()

```
Rcpp::stats::NormGenerator::NormGenerator (
    double mean_ = 0.0,
    double sd_ = 1.0 ) [inline]
```

Definition at line 31 of file rnorm.h.

### 6.449.3 Member Function Documentation

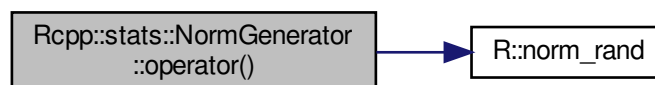
#### 6.449.3.1 operator>()

```
double Rcpp::stats::NormGenerator::operator() ( ) const [inline]
```

Definition at line 34 of file rnorm.h.

References `mean`, `R::norm_rand()`, and `sd`.

Here is the call graph for this function:





## 6.449.4 Member Data Documentation

### 6.449.4.1 mean

```
double Rcpp::stats::NormGenerator::mean [private]
```

Definition at line 39 of file rnorm.h.

Referenced by operator().

### 6.449.4.2 sd

```
double Rcpp::stats::NormGenerator::sd [private]
```

Definition at line 40 of file rnorm.h.

Referenced by operator().

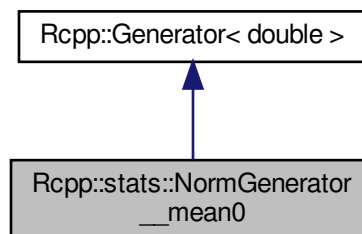
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rnorm.h](#)

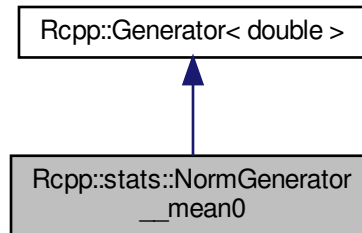
## 6.450 Rcpp::stats::NormGenerator\_\_mean0 Class Reference

```
#include <rnorm.h>
```

Inheritance diagram for Rcpp::stats::NormGenerator\_\_mean0:



Collaboration diagram for Rcpp::stats::NormGenerator\_\_mean0:



## Public Member Functions

- [NormGenerator\\_\\_mean0](#) (double sd\_=1.0)
- double [operator\(\)](#) () const

## Private Attributes

- double [sd](#)

## Additional Inherited Members

### 6.450.1 Detailed Description

Definition at line 56 of file `rnorm.h`.

### 6.450.2 Constructor & Destructor Documentation

#### 6.450.2.1 NormGenerator\_\_mean0()

```
Rcpp::stats::NormGenerator__mean0::NormGenerator__mean0 (  
    double sd_ = 1.0 ) [inline]
```

Definition at line 59 of file `rnorm.h`.

### 6.450.3 Member Function Documentation

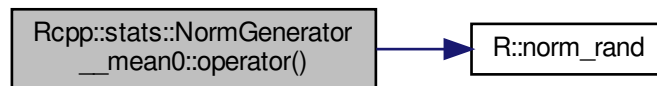
#### 6.450.3.1 operator()

```
double Rcpp::stats::NormGenerator__mean0::operator() ( ) const [inline]
```

Definition at line 61 of file rnorm.h.

References `R::norm_rand()`, and `sd`.

Here is the call graph for this function:



### 6.450.4 Member Data Documentation

#### 6.450.4.1 sd

```
double Rcpp::stats::NormGenerator__mean0::sd [private]
```

Definition at line 66 of file rnorm.h.

Referenced by `operator()`.

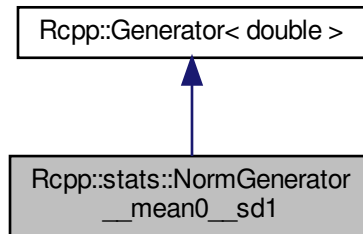
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rnorm.h](#)

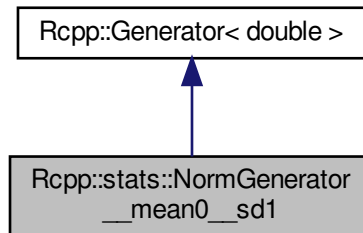
## 6.451 Rcpp::stats::NormGenerator\_\_mean0\_\_sd1 Class Reference

```
#include <rnorm.h>
```

Inheritance diagram for Rcpp::stats::NormGenerator\_\_mean0\_\_sd1:



Collaboration diagram for Rcpp::stats::NormGenerator\_\_mean0\_\_sd1:



### Public Member Functions

- [NormGenerator\\_\\_mean0\\_\\_sd1 \(\)](#)
- `double operator() () const`

### Additional Inherited Members

#### 6.451.1 Detailed Description

Definition at line 69 of file rnorm.h.

## 6.451.2 Constructor & Destructor Documentation

### 6.451.2.1 NormGenerator\_\_mean0\_\_sd1()

```
Rcpp::stats::NormGenerator__mean0__sd1::NormGenerator__mean0__sd1 ( ) [inline]
```

Definition at line 72 of file rnorm.h.

## 6.451.3 Member Function Documentation

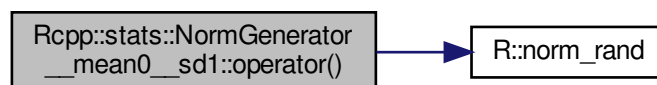
### 6.451.3.1 operator>()

```
double Rcpp::stats::NormGenerator__mean0__sd1::operator() ( ) const [inline]
```

Definition at line 74 of file rnorm.h.

References [R::norm\\_rand\(\)](#).

Here is the call graph for this function:



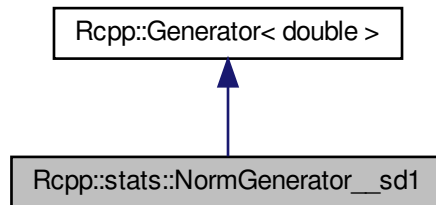
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rnorm.h](#)

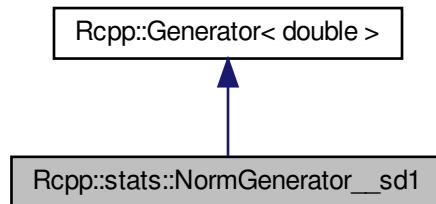
## 6.452 Rcpp::stats::NormGenerator\_\_sd1 Class Reference

```
#include <rnorm.h>
```

Inheritance diagram for Rcpp::stats::NormGenerator\_\_sd1:



Collaboration diagram for Rcpp::stats::NormGenerator\_\_sd1:



### Public Member Functions

- [NormGenerator\\_\\_sd1](#) (double mean\_=0.0)
- double [operator\(\)](#) () const

### Private Attributes

- double [mean](#)

## Additional Inherited Members

### 6.452.1 Detailed Description

Definition at line 43 of file rnorm.h.

### 6.452.2 Constructor & Destructor Documentation

#### 6.452.2.1 NormGenerator\_\_sd1()

```
Rcpp::stats::NormGenerator__sd1::NormGenerator__sd1 (  
    double mean_ = 0.0 ) [inline]
```

Definition at line 46 of file rnorm.h.

### 6.452.3 Member Function Documentation

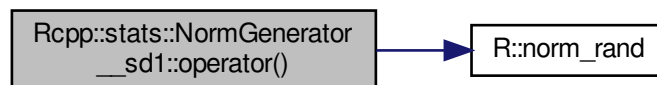
#### 6.452.3.1 operator>()

```
double Rcpp::stats::NormGenerator__sd1::operator() ( ) const [inline]
```

Definition at line 48 of file rnorm.h.

References `mean`, and `R::norm_rand()`.

Here is the call graph for this function:



## 6.452.4 Member Data Documentation

### 6.452.4.1 mean

```
double Rcpp::stats::NormGenerator__sd1::mean [private]
```

Definition at line 53 of file rnorm.h.

Referenced by operator().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/[rnorm.h](#)

## 6.453 Rcpp::sugar::not\_ < RTYPE, NA > Class Template Reference

```
#include <not.h>
```

### Public Types

- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)

### Public Member Functions

- int [apply](#) ([STORAGE](#) x) const

### 6.453.1 Detailed Description

```
template<int RTYPE, bool NA>  
class Rcpp::sugar::not_ < RTYPE, NA >
```

Definition at line 29 of file not.h.

### 6.453.2 Member Typedef Documentation



### 6.453.2.1 STORAGE

```
template<int RTYPE, bool NA>
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::not_< RTYPE, NA >::STORAGE
```

Definition at line 31 of file not.h.

## 6.453.3 Member Function Documentation

### 6.453.3.1 apply()

```
template<int RTYPE, bool NA>
int Rcpp::sugar::not_< RTYPE, NA >::apply (
    STORAGE x ) const [inline]
```

Definition at line 32 of file not.h.

Referenced by Rcpp::sugar::Not\_Vector< RTYPE, NA, T >::operator[]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/not.h

## 6.454 Rcpp::sugar::not\_< CPLXSP, false > Class Reference

```
#include <not.h>
```

### Public Member Functions

- int [apply](#) (Rcomplex x) const

### 6.454.1 Detailed Description

Definition at line 66 of file not.h.

### 6.454.2 Member Function Documentation

### 6.454.2.1 `apply()`

```
int Rcpp::sugar::not_< CPLXSCP, false >::apply (
    Rcomplex x ) const [inline]
```

Definition at line 68 of file not.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/not.h](#)

## 6.455 `Rcpp::sugar::not_< CPLXSCP, NA >` Class Template Reference

```
#include <not.h>
```

### Public Member Functions

- int [apply](#) (Rcomplex x) const

### 6.455.1 Detailed Description

```
template<bool NA>
class Rcpp::sugar::not_< CPLXSCP, NA >
```

Definition at line 59 of file not.h.

### 6.455.2 Member Function Documentation

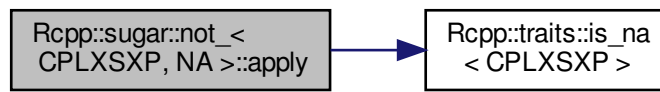
### 6.455.2.1 apply()

```
template<bool NA>
int Rcpp::sugar::not_< CPLXSXP, NA >::apply (
    Rcomplex x ) const [inline]
```

Definition at line 61 of file not.h.

References Rcpp::traits::is\_na< CPLXSXP >().

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/not.h](#)

## 6.456 Rcpp::sugar::not\_< REALSXP, false > Class Reference

```
#include <not.h>
```

### Public Member Functions

- int [apply](#) (double x) const

### 6.456.1 Detailed Description

Definition at line 52 of file not.h.

### 6.456.2 Member Function Documentation

### 6.456.2.1 `apply()`

```
int Rcpp::sugar::not_< REALSXP, false >::apply (
    double x ) const [inline]
```

Definition at line 54 of file `not.h`.

The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/operators/not.h`

## 6.457 `Rcpp::sugar::not_< REALSXP, NA >` Class Template Reference

```
#include <not.h>
```

### Public Member Functions

- `int apply` (double x) const

### 6.457.1 Detailed Description

```
template<bool NA>
class Rcpp::sugar::not_< REALSXP, NA >
```

Definition at line 45 of file `not.h`.

### 6.457.2 Member Function Documentation

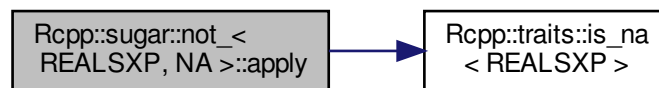
### 6.457.2.1 apply()

```
template<bool NA>
int Rcpp::sugar::not_< REALSXP, NA >::apply (
    double x ) const [inline]
```

Definition at line 47 of file not.h.

References Rcpp::traits::is\_na< REALSXP >().

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/not.h](#)

## 6.458 Rcpp::sugar::not\_< RTYPE, false > Class Template Reference

```
#include <not.h>
```

### Public Types

- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) STORAGE

### Public Member Functions

- int [apply](#) (STORAGE x) const

### 6.458.1 Detailed Description

```
template<int RTYPE>
class Rcpp::sugar::not_< RTYPE, false >
```

Definition at line 37 of file not.h.

## 6.458.2 Member Typedef Documentation

### 6.458.2.1 STORAGE

```
template<int RTYPE>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::not_< RTYPE, false >::STORAGE
```

Definition at line 39 of file not.h.

## 6.458.3 Member Function Documentation

### 6.458.3.1 apply()

```
template<int RTYPE>
int Rcpp::sugar::not_< RTYPE, false >::apply (
    STORAGE x ) const [inline]
```

Definition at line 40 of file not.h.

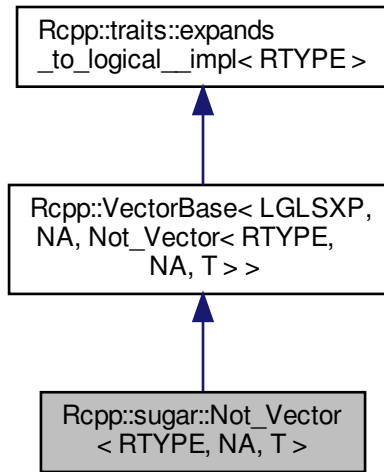
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/not.h](#)

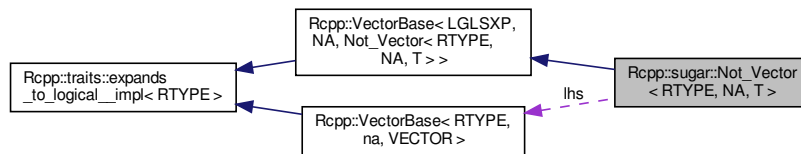
## 6.459 Rcpp::sugar::Not\_Vector< RTYPE, NA, T > Class Template Reference

```
#include <not.h>
```

Inheritance diagram for Rcpp::sugar::Not\_Vector< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Not\_Vector< RTYPE, NA, T >:



## Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `not_< RTYPE, NA >` `OPERATOR`

## Public Member Functions

- `Not_Vector` (const `VEC_TYPE` &lhs\_)
- `STORAGE operator[]` (R\_xlen\_t i) const
- `R_xlen_t size` () const

## Private Attributes

- const [VEC\\_TYPE](#) & lhs
- [OPERATOR](#) op

### 6.459.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::Not_Vector< RTYPE, NA, T >
```

Definition at line 76 of file not.h.

### 6.459.2 Member Typedef Documentation

#### 6.459.2.1 OPERATOR

```
template<int RTYPE, bool NA, typename T >  
typedef not\_<RTYPE,NA> Rcpp::sugar::Not\_Vector< RTYPE, NA, T >::OPERATOR
```

Definition at line 80 of file not.h.

#### 6.459.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >  
typedef traits::storage\_type<RTYPE>::type Rcpp::sugar::Not\_Vector< RTYPE, NA, T >::STORAGE
```

Definition at line 79 of file not.h.

#### 6.459.2.3 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Not\_Vector< RTYPE, NA, T >::VEC\_TYPE
```

Definition at line 78 of file not.h.

### 6.459.3 Constructor & Destructor Documentation



### 6.459.3.1 Not\_Vector()

```
template<int RTYPE, bool NA, typename T >  
Rcpp::sugar::Not_Vector< RTYPE, NA, T >::Not_Vector (  
    const VEC_TYPE & lhs_ ) [inline]
```

Definition at line 82 of file not.h.

## 6.459.4 Member Function Documentation

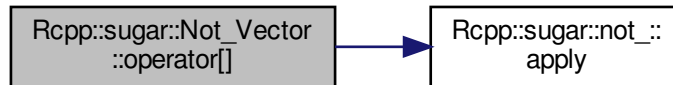
### 6.459.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >  
STORAGE Rcpp::sugar::Not_Vector< RTYPE, NA, T >::operator[] (  
    R_xlen_t i ) const [inline]
```

Definition at line 85 of file not.h.

References [Rcpp::sugar::not\\_< RTYPE, NA >::apply\(\)](#), [Rcpp::sugar::Not\\_Vector< RTYPE, NA, T >::lhs](#), and [Rcpp::sugar::Not\\_Vector< RTYPE, NA, T >::op](#).

Here is the call graph for this function:



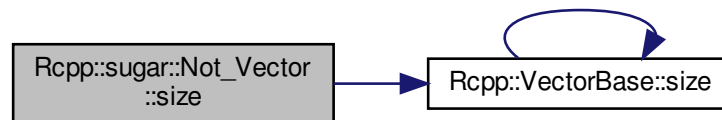
### 6.459.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Not_Vector< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 89 of file not.h.

References `Rcpp::sugar::Not_Vector< RTYPE, NA, T >::lhs`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`.

Here is the call graph for this function:



## 6.459.5 Member Data Documentation

### 6.459.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Not_Vector< RTYPE, NA, T >::lhs [private]
```

Definition at line 92 of file not.h.

Referenced by `Rcpp::sugar::Not_Vector< RTYPE, NA, T >::operator[]()`, and `Rcpp::sugar::Not_Vector< RTYPE, NA, T >::size()`.

### 6.459.5.2 op

```
template<int RTYPE, bool NA, typename T >
OPERATOR Rcpp::sugar::Not_Vector< RTYPE, NA, T >::op [private]
```

Definition at line 93 of file not.h.

Referenced by `Rcpp::sugar::Not_Vector< RTYPE, NA, T >::operator[]()`.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/not.h](#)

## 6.460 Rcpp::Nullable< T > Class Template Reference

```
#include <Nullable.h>
```

### Public Member Functions

- [Nullable](#) ()
- [Nullable](#) (const T &t)
- [Nullable](#) (SEXP t)
- [Nullable](#) & [operator=](#) (SEXP SEXP)
- [operator SEXP](#) () const
- SEXP [get](#) () const
- bool [isUsable](#) () const
- bool [isNull](#) () const
- bool [isNotNull](#) () const
- bool [isSet](#) (void) const
- T [as](#) ()
- T [clone](#) () const

### Private Member Functions

- void [checkIfSet](#) (void) const

### Private Attributes

- SEXP [m\\_Sexp](#)
- bool [m\\_set](#)

### 6.460.1 Detailed Description

```
template<class T>  
class Rcpp::Nullable< T >
```

Definition at line 33 of file Nullable.h.

### 6.460.2 Constructor & Destructor Documentation

### 6.460.2.1 Nullable() [1/3]

```
template<class T >  
Rcpp::Nullable< T >::Nullable ( ) [inline]
```

Empty no-argument constructor of a [Nullable](#) object

Assigns (R's) NULL value, and sets validator to FALSE

Definition at line 41 of file Nullable.h.

### 6.460.2.2 Nullable() [2/3]

```
template<class T >  
Rcpp::Nullable< T >::Nullable (   
    const T & t ) [inline]
```

Template constructor of a [Nullable](#) object

Assigns object, and set validator to TRUE

Definition at line 49 of file Nullable.h.

### 6.460.2.3 Nullable() [3/3]

```
template<class T >  
Rcpp::Nullable< T >::Nullable (   
    SEXP t ) [inline]
```

Standard constructor of a [Nullable](#) object

#### Parameters

<i>SEXP</i>	is stored
-------------	-----------

Definition at line 56 of file Nullable.h.

References `Rcpp::Nullable< T >::m_set`, and `Rcpp::Nullable< T >::m_sexp`.

## 6.460.3 Member Function Documentation

### 6.460.3.1 as()

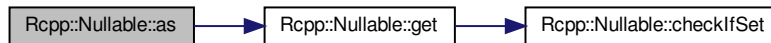
```
template<class T >
T Rcpp::Nullable< T >::as ( ) [inline]
```

Returns m\_sexp as a T

Definition at line 129 of file Nullable.h.

References Rcpp::Nullable< T >::get().

Here is the call graph for this function:



### 6.460.3.2 checkIfSet()

```
template<class T >
void Rcpp::Nullable< T >::checkIfSet (
    void ) const [inline], [private]
```

Definition at line 140 of file Nullable.h.

References Rcpp::Nullable< T >::m\_set.

Referenced by Rcpp::Nullable< T >::get(), Rcpp::Nullable< T >::isNull(), and Rcpp::Nullable< T >::operator SEXP().

### 6.460.3.3 clone()

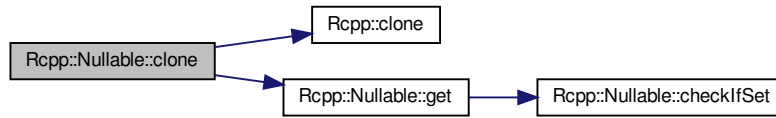
```
template<class T >
T Rcpp::Nullable< T >::clone ( ) const [inline]
```

Return a clone of m\_sexp as a T

Definition at line 134 of file Nullable.h.

References Rcpp::clone(), and Rcpp::Nullable< T >::get().

Here is the call graph for this function:



#### 6.460.3.4 get()

```

template<class T >
SEXPR Rcpp::Nullable< T >::get ( ) const [inline]
  
```

[get\(\)](#) accessor for object

#### Exceptions

'not	initialized' if object has not been set
------	---

Definition at line 89 of file Nullable.h.

References `Rcpp::Nullable< T >::checkIfSet()`, and `Rcpp::Nullable< T >::m_sexp`.

Referenced by `Rcpp::Nullable< T >::as()`, and `Rcpp::Nullable< T >::clone()`.

Here is the call graph for this function:



#### 6.460.3.5 isNotNull()

```

template<class T >
bool Rcpp::Nullable< T >::isNotNull ( ) const [inline]
  
```

Boolean test for not NULL

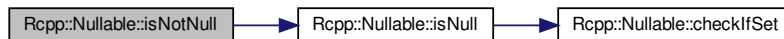
**Exceptions**

'not	initialized' if object has not been set
------	---

Definition at line 116 of file Nullable.h.

References Rcpp::Nullable< T >::isNull().

Here is the call graph for this function:

**6.460.3.6 isNull()**

```
template<class T >  
bool Rcpp::Nullable< T >::isNull ( ) const [inline]
```

Boolean test for NULL

**Exceptions**

'not	initialized' if object has not been set
------	---

Definition at line 106 of file Nullable.h.

References Rcpp::Nullable< T >::checkIfSet(), and Rcpp::Nullable< T >::m\_sexp.

Referenced by Rcpp::Nullable< T >::isNotNull().

Here is the call graph for this function:



### 6.460.3.7 isSet()

```
template<class T >
bool Rcpp::Nullable< T >::isSet (
    void ) const [inline]
```

Test function to check if object has been initialized

Definition at line 124 of file Nullable.h.

References Rcpp::Nullable< T >::m\_set.

### 6.460.3.8 isUsable()

```
template<class T >
bool Rcpp::Nullable< T >::isUsable ( ) const [inline]
```

Boolean test for usability as a T

Definition at line 97 of file Nullable.h.

References Rcpp::Nullable< T >::m\_set, and Rcpp::Nullable< T >::m\_sexp.

### 6.460.3.9 operator SEXP()

```
template<class T >
Rcpp::Nullable< T >::operator SEXP ( ) const [inline]
```

operator SEXP() to return nullable object

#### Exceptions

'not initialized' if object has not been set
--

Definition at line 79 of file Nullable.h.

References Rcpp::Nullable< T >::checkIfSet(), and Rcpp::Nullable< T >::m\_sexp.



Here is the call graph for this function:



### 6.460.3.10 operator=( )

```
template<class T >  
Nullable& Rcpp::Nullable< T >::operator= (  
    SEXP sexp ) [inline]
```

Copy constructor for [Nullable](#) object

#### Parameters

<i>SEXP</i>	is used to update internal copy
-------------	---------------------------------

Definition at line 68 of file Nullable.h.

References [Rcpp::Nullable< T >::m\\_set](#), and [Rcpp::Nullable< T >::m\\_sexp](#).

## 6.460.4 Member Data Documentation

### 6.460.4.1 m\_set

```
template<class T >  
bool Rcpp::Nullable< T >::m_set [private]
```

Definition at line 138 of file Nullable.h.

Referenced by [Rcpp::Nullable< T >::checkIfSet\(\)](#), [Rcpp::Nullable< T >::isSet\(\)](#), [Rcpp::Nullable< T >::isUsable\(\)](#), [Rcpp::Nullable< T >::Nullable\(\)](#), and [Rcpp::Nullable< T >::operator=\(\)](#).

### 6.460.4.2 m\_sexp

```
template<class T >
SEXP Rcpp::Nullable< T >::m_sexp [private]
```

Definition at line 137 of file Nullable.h.

Referenced by Rcpp::Nullable< T >::get(), Rcpp::Nullable< T >::isNull(), Rcpp::Nullable< T >::isUsable(), Rcpp::Nullable< T >::Nullable(), Rcpp::Nullable< T >::operator SEXP(), and Rcpp::Nullable< T >::operator=().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Nullable.h](#)

## 6.461 Rcpp::traits::num2type< N > Struct Template Reference

```
#include <num2type.h>
```

### 6.461.1 Detailed Description

```
template<std::size_t N>
struct Rcpp::traits::num2type< N >
```

Definition at line 29 of file num2type.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/num2type.h](#)

## 6.462 Rcpp::object< T > Class Template Reference

```
#include <Module.h>
```

### Public Types

- typedef T [object\\_type](#)

### Public Member Functions

- [object](#) (T \*ptr\_)
- [operator T\\*](#) ()
- T \* [operator->](#) ()
- T & [operator&](#) ()

## Public Attributes

- `T * ptr`

### 6.462.1 Detailed Description

```
template<typename T>  
class Rcpp::object< T >
```

Definition at line 65 of file Module.h.

### 6.462.2 Member Typedef Documentation

#### 6.462.2.1 object\_type

```
template<typename T >  
typedef T Rcpp::object< T >::object_type
```

Definition at line 67 of file Module.h.

### 6.462.3 Constructor & Destructor Documentation

#### 6.462.3.1 object()

```
template<typename T >  
Rcpp::object< T >::object (  
    T * ptr_ ) [inline]
```

Definition at line 68 of file Module.h.

### 6.462.4 Member Function Documentation

#### 6.462.4.1 operator T\*()

```
template<typename T >  
Rcpp::object< T >::operator T* ( ) [inline]
```

Definition at line 69 of file Module.h.

References Rcpp::object< T >::ptr.

#### 6.462.4.2 operator&()

```
template<typename T >  
T& Rcpp::object< T >::operator& ( ) [inline]
```

Definition at line 71 of file Module.h.

References Rcpp::object< T >::ptr.

#### 6.462.4.3 operator->()

```
template<typename T >  
T* Rcpp::object< T >::operator-> ( ) [inline]
```

Definition at line 70 of file Module.h.

References Rcpp::object< T >::ptr.

### 6.462.5 Member Data Documentation

#### 6.462.5.1 ptr

```
template<typename T >  
T* Rcpp::object< T >::ptr
```

Definition at line 72 of file Module.h.

Referenced by Rcpp::object< T >::operator T\*(), Rcpp::object< T >::operator&(), and Rcpp::object< T >::operator->().

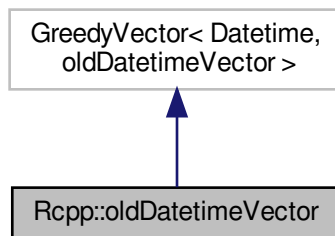
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Module.h](#)

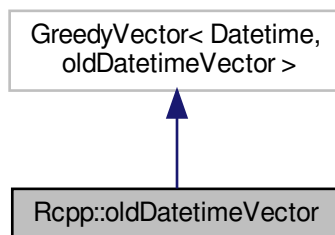
## 6.463 Rcpp::oldDatetimeVector Class Reference

```
#include <oldDatetimeVector.h>
```

Inheritance diagram for Rcpp::oldDatetimeVector:



Collaboration diagram for Rcpp::oldDatetimeVector:



### Public Member Functions

- [oldDatetimeVector](#) (SEXP vec)
- [oldDatetimeVector](#) (int n)
- `std::vector< Datetime > getDatetimes () const`

### 6.463.1 Detailed Description

Definition at line 31 of file `oldDatetimeVector.h`.

## 6.463.2 Constructor & Destructor Documentation

### 6.463.2.1 oldDatetimeVector() [1/2]

```
Rcpp::oldDatetimeVector::oldDatetimeVector (  
    SEXP vec ) [inline]
```

Definition at line 33 of file oldDatetimeVector.h.

### 6.463.2.2 oldDatetimeVector() [2/2]

```
Rcpp::oldDatetimeVector::oldDatetimeVector (  
    int n ) [inline]
```

Definition at line 34 of file oldDatetimeVector.h.

## 6.463.3 Member Function Documentation

### 6.463.3.1 getDatetimes()

```
std::vector<Datetime> Rcpp::oldDatetimeVector::getDatetimes ( ) const [inline]
```

Definition at line 36 of file oldDatetimeVector.h.

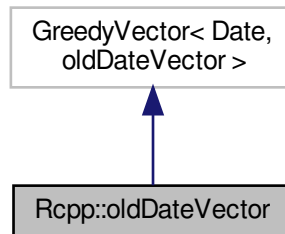
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/date\\_datetime/oldDatetimeVector.h](#)

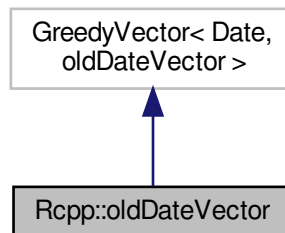
## 6.464 Rcpp::oldDateVector Class Reference

```
#include <oldDateVector.h>
```

Inheritance diagram for Rcpp::oldDateVector:



Collaboration diagram for Rcpp::oldDateVector:



### Public Member Functions

- [oldDateVector](#) (SEXP vec)
- [oldDateVector](#) (int n)
- `std::vector< Date > getDates () const`

### 6.464.1 Detailed Description

Definition at line 30 of file `oldDateVector.h`.

## 6.464.2 Constructor & Destructor Documentation

### 6.464.2.1 `oldDateVector()` [1/2]

```
Rcpp::oldDateVector::oldDateVector (
    SEXP vec ) [inline]
```

Definition at line 32 of file `oldDateVector.h`.

### 6.464.2.2 `oldDateVector()` [2/2]

```
Rcpp::oldDateVector::oldDateVector (
    int n ) [inline]
```

Definition at line 33 of file `oldDateVector.h`.

## 6.464.3 Member Function Documentation

### 6.464.3.1 `getDates()`

```
std::vector<Date> Rcpp::oldDateVector::getDates ( ) const [inline]
```

Definition at line 35 of file `oldDateVector.h`.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/date\\_datetime/oldDateVector.h](#)

## 6.465 `Rcpp::traits::one_type< T >` Class Template Reference

```
#include <one_type.h>
```

### Public Member Functions

- [operator T \(\)](#) const



## Private Member Functions

- Rcomplex [op \(true\\_type\)](#) const
- T [op \(false\\_type\)](#) const

### 6.465.1 Detailed Description

```
template<typename T>  
class Rcpp::traits::one_type< T >
```

Definition at line 35 of file one\_type.h.

### 6.465.2 Member Function Documentation

#### 6.465.2.1 [op\(\)](#) [1/2]

```
template<typename T >  
T Rcpp::traits::one_type< T >::op (  
    false_type ) const [inline], [private]
```

Definition at line 44 of file one\_type.h.

#### 6.465.2.2 [op\(\)](#) [2/2]

```
template<typename T >  
Rcomplex Rcpp::traits::one_type< T >::op (  
    true_type ) const [inline], [private]
```

Definition at line 37 of file one\_type.h.

Referenced by [Rcpp::traits::one\\_type< T >::operator T\(\)](#).

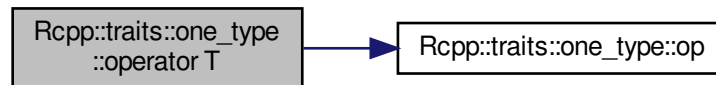
### 6.465.2.3 operator T()

```
template<typename T >
Rcpp::traits::one_type< T >::operator T ( ) const [inline]
```

Definition at line 49 of file one\_type.h.

References Rcpp::traits::one\_type< T >::op().

Here is the call graph for this function:



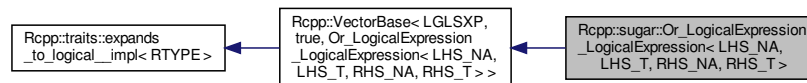
The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/one\_type.h

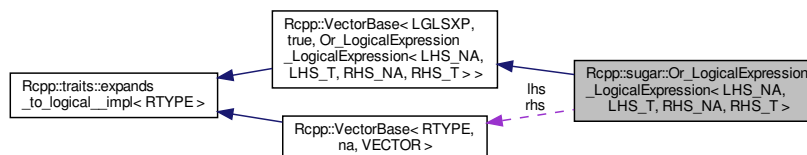
## 6.466 Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< LGLSXP, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)

## Public Member Functions

- [Or\\_LogicalExpression\\_LogicalExpression](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- int [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_TYPE](#) &lhs
- const [RHS\\_TYPE](#) &rhs

### 6.466.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>  
class Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 202 of file or.h.

### 6.466.2 Member Typedef Documentation

#### 6.466.2.1 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::VectorBase<LGLSXP, LHS_NA, LHS_T> Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression<  
LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS\_TYPE
```

Definition at line 204 of file or.h.

#### 6.466.2.2 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::VectorBase<LGLSXP, RHS_NA, RHS_T> Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression<  
LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS\_TYPE
```

Definition at line 205 of file or.h.

## 6.466.3 Constructor & Destructor Documentation

### 6.466.3.1 Or\_LogicalExpression\_LogicalExpression()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::Or_LogicalExpression_LogicalExpression
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 207 of file or.h.

## 6.466.4 Member Function Documentation

### 6.466.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
int Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[]
(
    R_xlen_t i ) const [inline]
```

Definition at line 209 of file or.h.

References `Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

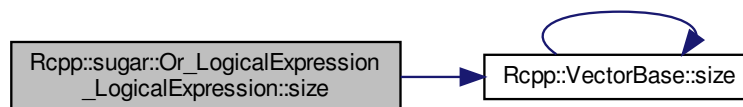
### 6.466.4.2 size()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >↔
::size ( ) const [inline]
```

Definition at line 214 of file or.h.

References `Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`.

Here is the call graph for this function:



## 6.466.5 Member Data Documentation

### 6.466.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
const LHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T  
>::lhs [private]
```

Definition at line 217 of file or.h.

Referenced by Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, false, RHS\_T >::operator[](), Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, false, RHS\_T >::operator[](), Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, false, RHS\_T >::size(), and Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, false, RHS\_T >::size().

### 6.466.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
const RHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T  
>::rhs [private]
```

Definition at line 218 of file or.h.

Referenced by Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, false, RHS\_T >::operator[](), and Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, false, RHS\_T >::operator[]().

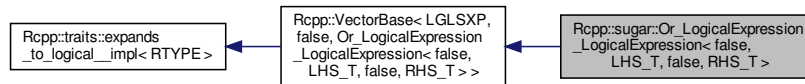
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

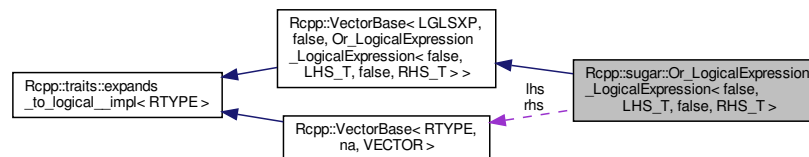
## 6.467 Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< LGLSXP, false, RHS\_T > [RHS\\_TYPE](#)

### Public Member Functions

- [Or\\_LogicalExpression\\_LogicalExpression](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- int [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

### Private Attributes

- const [LHS\\_TYPE](#) &lhs
- const [RHS\\_TYPE](#) &rhs

## 6.467.1 Detailed Description

```
template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >
```

Definition at line 261 of file or.h.

## 6.467.2 Member Typedef Documentation

### 6.467.2.1 LHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP, false, LHS_T> Rcpp::sugar::Or_LogicalExpression_LogicalExpression<
false, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 264 of file or.h.

### 6.467.2.2 RHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP, false, RHS_T> Rcpp::sugar::Or_LogicalExpression_LogicalExpression<
false, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 265 of file or.h.

## 6.467.3 Constructor & Destructor Documentation

### 6.467.3.1 Or\_LogicalExpression\_LogicalExpression()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::Or_LogicalExpression_LogicalExpE
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 267 of file or.h.

## 6.467.4 Member Function Documentation

### 6.467.4.1 operator[]()

```
template<typename LHS_T , typename RHS_T >
int Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::operator[]
(
    R_xlen_t i ) const [inline]
```

Definition at line 269 of file or.h.

References `Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

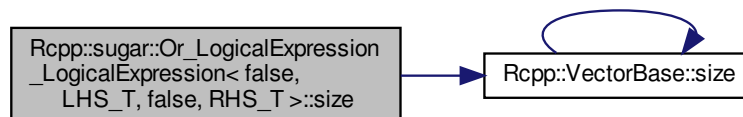
### 6.467.4.2 size()

```
template<typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >::size
( ) const [inline]
```

Definition at line 273 of file or.h.

References `Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`.

Here is the call graph for this function:



## 6.467.5 Member Data Documentation



6.467.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T
>::lhs [private]
```

Definition at line 276 of file or.h.

6.467.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T
>::rhs [private]
```

Definition at line 277 of file or.h.

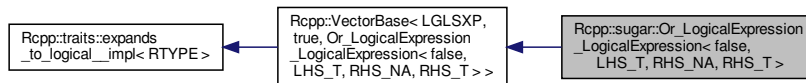
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

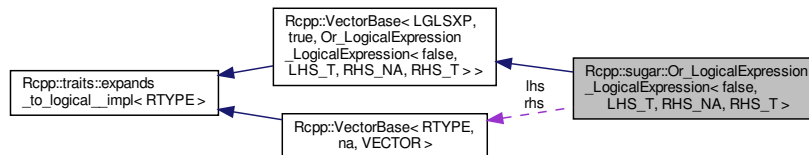
6.468 Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, RHS\_NA, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< LGLSXP, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)

## Public Member Functions

- [Or\\_LogicalExpression\\_LogicalExpression](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- int [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_TYPE](#) & lhs
- const [RHS\\_TYPE](#) & rhs

### 6.468.1 Detailed Description

```
template<typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >
```

Definition at line 221 of file or.h.

### 6.468.2 Member Typedef Documentation

#### 6.468.2.1 LHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP, false, LHS_T> Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression<
false, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 224 of file or.h.

#### 6.468.2.2 RHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<LGLSXP, RHS_NA, RHS_T> Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression<
false, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 225 of file or.h.

## 6.468.3 Constructor & Destructor Documentation

### 6.468.3.1 Or\_LogicalExpression\_LogicalExpression()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::Or_LogicalExpression_LogicalExpression  
(  
    const LHS_TYPE & lhs_,  
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 227 of file or.h.

## 6.468.4 Member Function Documentation

### 6.468.4.1 operator[]()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
int Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::operator[]  
(  
    R_xlen_t i ) const [inline]
```

Definition at line 229 of file or.h.

References Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.468.4.2 size()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
R_xlen_t Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >::size  
( ) const [inline]
```

Definition at line 234 of file or.h.

References Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



## 6.468.5 Member Data Documentation

### 6.468.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T
>::lhs [private]
```

Definition at line 237 of file or.h.

### 6.468.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T
>::rhs [private]
```

Definition at line 238 of file or.h.

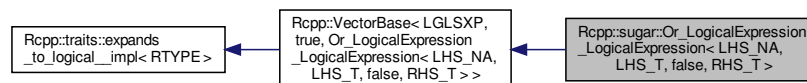
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

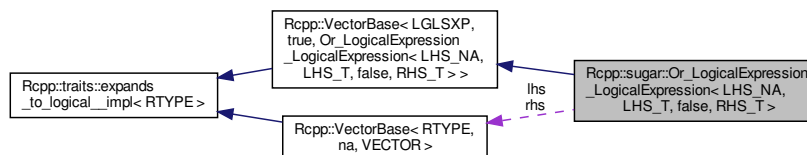
## 6.469 Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, false, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< LGLSXP, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< LGLSXP, false, RHS\_T > [RHS\\_TYPE](#)

## Public Member Functions

- [Or\\_LogicalExpression\\_LogicalExpression](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- int [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_TYPE](#) & lhs
- const [RHS\\_TYPE](#) & rhs

### 6.469.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, typename RHS_T>  
class Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >
```

Definition at line 241 of file or.h.

### 6.469.2 Member Typedef Documentation

#### 6.469.2.1 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef Rcpp::VectorBase<LGLSXP, LHS_NA, LHS_T> Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression<  
LHS_NA, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 244 of file or.h.

#### 6.469.2.2 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef Rcpp::VectorBase<LGLSXP, false, RHS_T> Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression<  
LHS_NA, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 245 of file or.h.

### 6.469.3 Constructor & Destructor Documentation

#### 6.469.3.1 Or\_LogicalExpression\_LogicalExpression()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::Or_LogicalExpression_LogicalExpression
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 247 of file or.h.

### 6.469.4 Member Function Documentation

#### 6.469.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
int Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::operator[]
(
    R_xlen_t i ) const [inline]
```

Definition at line 249 of file or.h.

References `Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

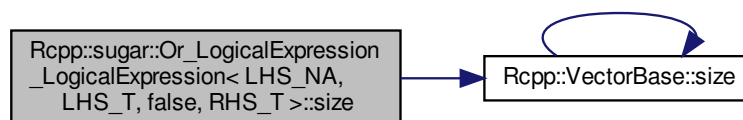
#### 6.469.4.2 size()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >::size
( ) const [inline]
```

Definition at line 254 of file or.h.

References `Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`.

Here is the call graph for this function:



## 6.469.5 Member Data Documentation

### 6.469.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T
>::lhs [private]
```

Definition at line 257 of file or.h.

### 6.469.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T
>::rhs [private]
```

Definition at line 258 of file or.h.

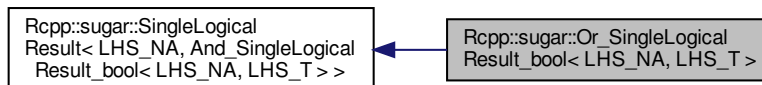
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

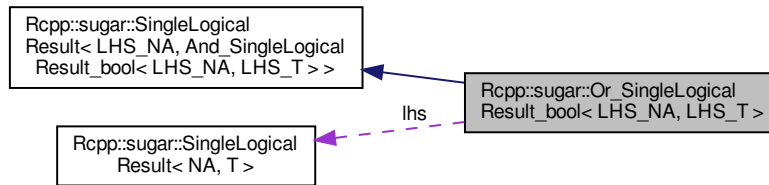
## 6.470 Rcpp::sugar::Or\_SingleLogicalResult\_bool< LHS\_NA, LHS\_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or\_SingleLogicalResult\_bool< LHS\_NA, LHS\_T >:



Collaboration diagram for `Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >`:



## Public Types

- typedef `SingleLogicalResult< LHS_NA, LHS_T > LHS_TYPE`
- typedef `SingleLogicalResult< LHS_NA, Or_SingleLogicalResult_bool< LHS_NA, LHS_T > > BASE`

## Public Member Functions

- `Or_SingleLogicalResult_bool` (const `LHS_TYPE` &lhs\_, bool rhs\_)
- void `apply` ()

## Private Attributes

- const `LHS_TYPE` &lhs
- bool rhs

## Additional Inherited Members

### 6.470.1 Detailed Description

```

template<bool LHS_NA, typename LHS_T>
class Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >
  
```

Definition at line 170 of file or.h.

### 6.470.2 Member Typedef Documentation



### 6.470.2.1 BASE

```
template<bool LHS_NA, typename LHS_T >
typedef SingleLogicalResult< LHS_NA , Or_SingleLogicalResult_bool<LHS_NA,LHS_T> > Rcpp::sugar::Or_SingleLogicalResult_bool<
LHS_NA, LHS_T >::BASE
```

Definition at line 181 of file or.h.

### 6.470.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T >
typedef SingleLogicalResult<LHS_NA,LHS_T> Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::LHS_TYPE
```

Definition at line 177 of file or.h.

## 6.470.3 Constructor & Destructor Documentation

### 6.470.3.1 Or\_SingleLogicalResult\_bool()

```
template<bool LHS_NA, typename LHS_T >
Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::Or_SingleLogicalResult_bool (
    const LHS_TYPE & lhs_,
    bool rhs_ ) [inline]
```

Definition at line 183 of file or.h.

## 6.470.4 Member Function Documentation

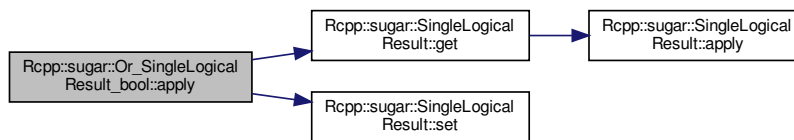
### 6.470.4.1 apply()

```
template<bool LHS_NA, typename LHS_T >
void Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply ( ) [inline]
```

Definition at line 186 of file or.h.

References `Rcpp::sugar::SingleLogicalResult< NA, T >::get()`, `Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::lhs`, `Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::rhs`, and `Rcpp::sugar::SingleLogicalResult< NA, T >::set()`.

Here is the call graph for this function:



## 6.470.5 Member Data Documentation

### 6.470.5.1 lhs

```
template<bool LHS_NA, typename LHS_T >
const LHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::lhs [private]
```

Definition at line 195 of file or.h.

Referenced by `Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply()`.

### 6.470.5.2 rhs

```
template<bool LHS_NA, typename LHS_T >
bool Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::rhs [private]
```

Definition at line 196 of file or.h.

Referenced by `Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply()`.

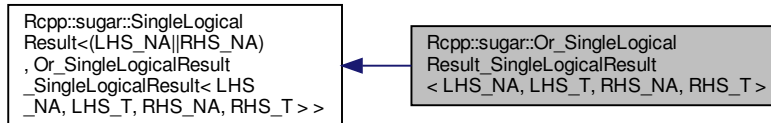
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/logical/or.h`

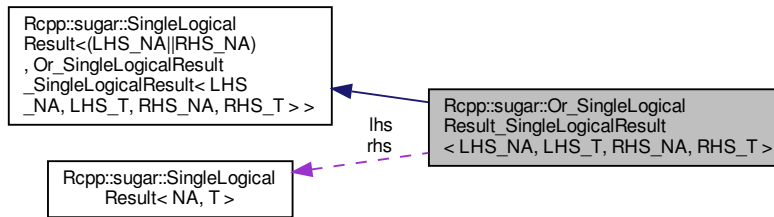
## 6.471 Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [SingleLogicalResult< LHS\\_NA, LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [SingleLogicalResult< RHS\\_NA, RHS\\_T >](#) [RHS\\_TYPE](#)
- typedef [SingleLogicalResult<\(LHS\\_NA||RHS\\_NA\), Or\\_SingleLogicalResult\\_SingleLogicalResult< LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >>](#) [BASE](#)

### Public Member Functions

- [Or\\_SingleLogicalResult\\_SingleLogicalResult](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- void [apply](#) ()

## Private Attributes

- const [LHS\\_TYPE](#) & lhs
- const [RHS\\_TYPE](#) & rhs

## Additional Inherited Members

### 6.471.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 29 of file or.h.

### 6.471.2 Member Typedef Documentation

#### 6.471.2.1 BASE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult< (LHS_NA || RHS_NA) , Or\_SingleLogicalResult\_SingleLogicalResult<LHS↔
_NA, LHS_T, RHS_NA, RHS_T> > Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS_NA, LHS_T,
RHS_NA, RHS_T >::BASE
```

Definition at line 41 of file or.h.

#### 6.471.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<LHS_NA, LHS_T> Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult<
LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS\_TYPE
```

Definition at line 36 of file or.h.

#### 6.471.2.3 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<RHS_NA, RHS_T> Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult<
LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS\_TYPE
```

Definition at line 37 of file or.h.

### 6.471.3 Constructor & Destructor Documentation

#### 6.471.3.1 Or\_SingleLogicalResult\_SingleLogicalResult()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::Or_SingleLogicalResult_S
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 43 of file or.h.

### 6.471.4 Member Function Documentation

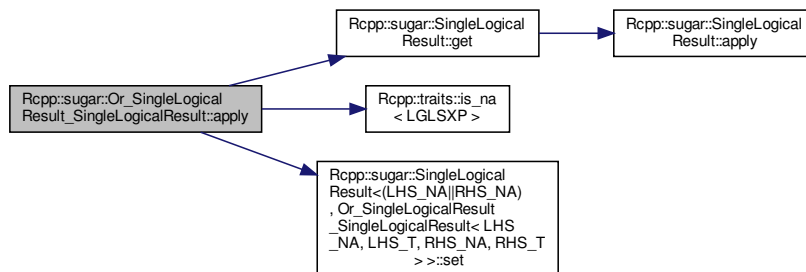
#### 6.471.4.1 apply()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
void Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::
::apply ( ) [inline]
```

Definition at line 46 of file or.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::traits::is\_na< LGLSXP >(), Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs, and Rcpp::sugar::SingleLogicalResult<(LHS\_NA||RHS\_NA), Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::set().

Here is the call graph for this function:



## 6.471.5 Member Data Documentation

### 6.471.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA,
RHS_T >::lhs [private]
```

Definition at line 58 of file or.h.

Referenced by `Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::apply()`, `Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::apply()`, `Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::apply()`, and `Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply()`.

### 6.471.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA,
RHS_T >::rhs [private]
```

Definition at line 59 of file or.h.

Referenced by `Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::apply()`, `Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::apply()`, `Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::apply()`, and `Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply()`.

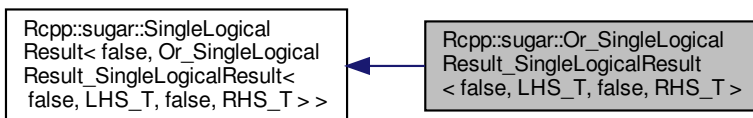
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/logical/or.h`

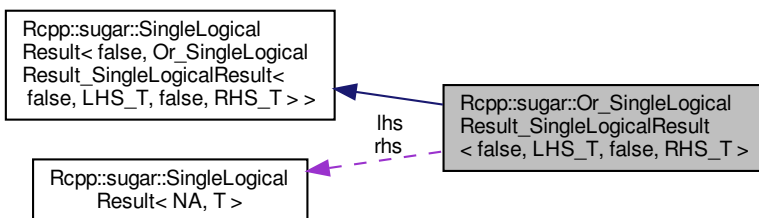
## 6.472 Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [SingleLogicalResult< false, LHS\\_T >](#) **LHS\_TYPE**
- typedef [SingleLogicalResult< false, RHS\\_T >](#) **RHS\_TYPE**
- typedef [SingleLogicalResult< false, Or\\_SingleLogicalResult\\_SingleLogicalResult< false, LHS\\_T, false, RHS\\_T > >](#) **BASE**

### Public Member Functions

- [Or\\_SingleLogicalResult\\_SingleLogicalResult](#) (const **LHS\_TYPE** &lhs\_, const **RHS\_TYPE** &rhs\_)
- void [apply](#) ()

### Private Attributes

- const **LHS\_TYPE** & lhs
- const **RHS\_TYPE** & rhs

## Additional Inherited Members

### 6.472.1 Detailed Description

```
template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >
```

Definition at line 136 of file or.h.

### 6.472.2 Member Typedef Documentation

#### 6.472.2.1 BASE

```
template<typename LHS_T , typename RHS_T >
typedef SingleLogicalResult< false, Or_SingleLogicalResult_SingleLogicalResult<false,LHS_T,false,RHS↔
_T> > Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >↔
::BASE
```

Definition at line 148 of file or.h.

#### 6.472.2.2 LHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef SingleLogicalResult<false,LHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 143 of file or.h.

#### 6.472.2.3 RHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef SingleLogicalResult<false,RHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 144 of file or.h.



## 6.472.3 Constructor & Destructor Documentation

### 6.472.3.1 Or\_SingleLogicalResult\_SingleLogicalResult()

```
template<typename LHS_T , typename RHS_T >  
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::Or_SingleLogicalResult_Si  
(  
    const LHS_TYPE & lhs_,  
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 150 of file or.h.

## 6.472.4 Member Function Documentation

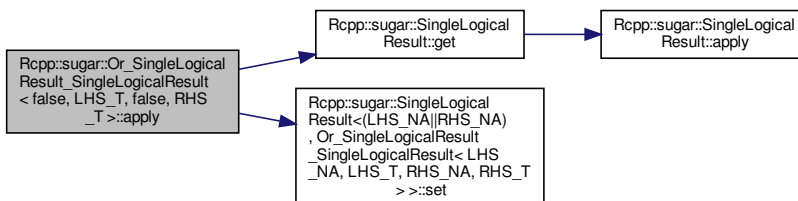
### 6.472.4.1 apply()

```
template<typename LHS_T , typename RHS_T >  
void Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >::apply  
( ) [inline]
```

Definition at line 153 of file or.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs, and Rcpp::sugar::SingleLogicalResult<(LHS\_NA||RHS\_NA), Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > >::set().

Here is the call graph for this function:



## 6.472.5 Member Data Documentation

### 6.472.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false,
RHS_T >::lhs [private]
```

Definition at line 163 of file or.h.

### 6.472.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false,
RHS_T >::rhs [private]
```

Definition at line 164 of file or.h.

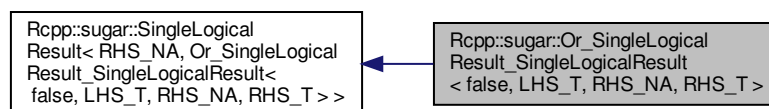
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

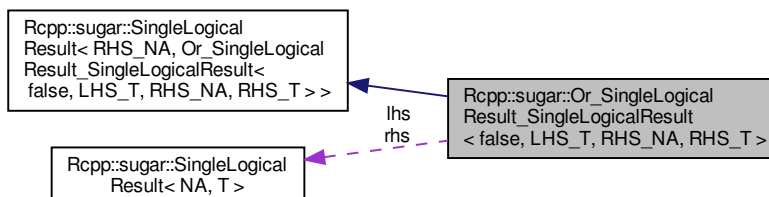
## 6.473 Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, RHS\_NA, RHS\_T >:



## Public Types

- typedef [SingleLogicalResult](#)< false, LHS\_T > [LHS\\_TYPE](#)
- typedef [SingleLogicalResult](#)< RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [SingleLogicalResult](#)< RHS\_NA, [Or\\_SingleLogicalResult\\_SingleLogicalResult](#)< false, LHS\_T, RHS\_NA, RHS\_T > > [BASE](#)

## Public Member Functions

- [Or\\_SingleLogicalResult\\_SingleLogicalResult](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- void [apply](#) ()

## Private Attributes

- const [LHS\\_TYPE](#) &lhs
- const [RHS\\_TYPE](#) &rhs

## Additional Inherited Members

### 6.473.1 Detailed Description

```
template<typename LHS_T, bool RHS_NA, typename RHS_T>  
class Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >
```

Definition at line 101 of file or.h.

### 6.473.2 Member Typedef Documentation

#### 6.473.2.1 BASE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef SingleLogicalResult< RHS_NA, Or\_SingleLogicalResult\_SingleLogicalResult<false, LHS_T, RHS↔  
_NA, RHS_T> > Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< false, LHS_T, RHS_NA, RHS↔  
_T >::BASE
```

Definition at line 113 of file or.h.

### 6.473.2.2 LHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<false,LHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 108 of file or.h.

### 6.473.2.3 RHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SingleLogicalResult<RHS_NA,RHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<
false, LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 109 of file or.h.

## 6.473.3 Constructor & Destructor Documentation

### 6.473.3.1 Or\_SingleLogicalResult\_SingleLogicalResult()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::Or_SingleLogicalResult_S
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 115 of file or.h.

## 6.473.4 Member Function Documentation

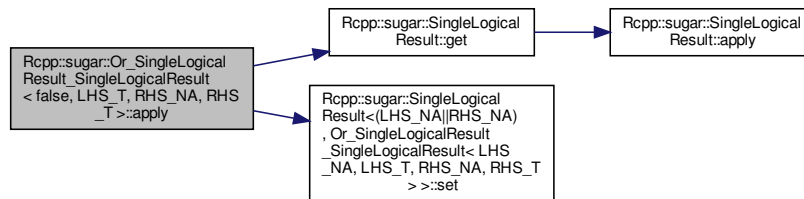
### 6.473.4.1 apply()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
void Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >::
::apply ( ) [inline]
```

Definition at line 118 of file or.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs, and Rcpp::sugar::SingleLogicalResult<(LHS\_NA||RHS\_NA), Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > >::set().

Here is the call graph for this function:



## 6.473.5 Member Data Documentation

### 6.473.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA,
RHS_T >::lhs [private]
```

Definition at line 129 of file or.h.

### 6.473.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA,
RHS_T >::rhs [private]
```

Definition at line 130 of file or.h.

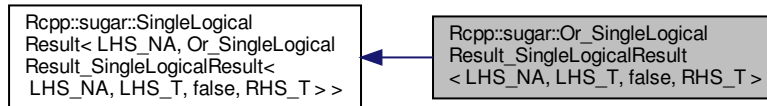
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

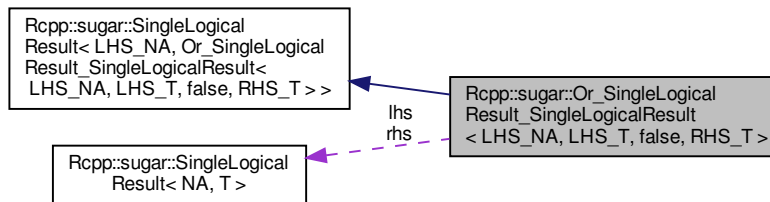
## 6.474 Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <or.h>
```

Inheritance diagram for Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [SingleLogicalResult< LHS\\_NA, LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [SingleLogicalResult< false, RHS\\_T >](#) [RHS\\_TYPE](#)
- typedef [SingleLogicalResult< LHS\\_NA, Or\\_SingleLogicalResult\\_SingleLogicalResult< LHS\\_NA, LHS\\_T, false, RHS\\_T >>](#) [BASE](#)

### Public Member Functions

- [Or\\_SingleLogicalResult\\_SingleLogicalResult](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- void [apply](#) ()

### Private Attributes

- const [LHS\\_TYPE](#) & [lhs](#)
- const [RHS\\_TYPE](#) & [rhs](#)

## Additional Inherited Members

### 6.474.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, typename RHS_T>  
class Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >
```

Definition at line 65 of file or.h.

### 6.474.2 Member Typedef Documentation

#### 6.474.2.1 BASE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef SingleLogicalResult< LHS_NA, Or_SingleLogicalResult_SingleLogicalResult<LHS_NA, LHS_T, false, RHS_T> > Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::BASE
```

Definition at line 77 of file or.h.

#### 6.474.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef SingleLogicalResult<LHS_NA, LHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 72 of file or.h.

#### 6.474.2.3 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef SingleLogicalResult<false, RHS_T> Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 73 of file or.h.

## 6.474.3 Constructor & Destructor Documentation

### 6.474.3.1 Or\_SingleLogicalResult\_SingleLogicalResult()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::Or_SingleLogicalResult_S
(
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 79 of file or.h.

## 6.474.4 Member Function Documentation

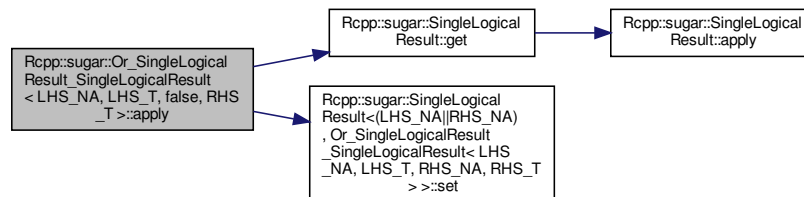
### 6.474.4.1 apply()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
void Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >::
::apply ( ) [inline]
```

Definition at line 82 of file or.h.

References `Rcpp::sugar::SingleLogicalResult< NA, T >::get()`, `Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, `Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`, and `Rcpp::sugar::SingleLogicalResult<(LHS_NA||RHS_NA), Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > >::set()`.

Here is the call graph for this function:



## 6.474.5 Member Data Documentation



### 6.474.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false,
RHS_T >::lhs [private]
```

Definition at line 93 of file or.h.

### 6.474.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_TYPE& Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false,
RHS_T >::rhs [private]
```

Definition at line 94 of file or.h.

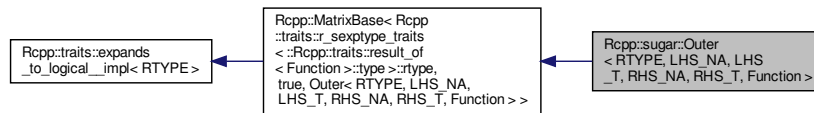
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/or.h

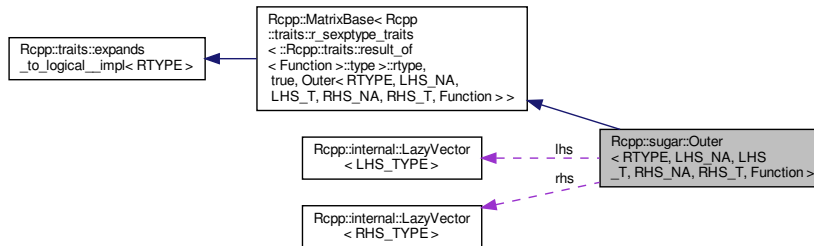
## 6.475 Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function > Class Template Reference

```
#include <outer.h>
```

Inheritance diagram for Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >:



Collaboration diagram for Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >:



## Public Types

- typedef `::Rcpp::traits::result_of< Function >::type` `result_type`
- typedef `Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T >` `LHS_TYPE`
- typedef `Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T >` `RHS_TYPE`
- typedef `Rcpp::internal::LazyVector< LHS_TYPE >` `LHS_LAZY`
- typedef `Rcpp::internal::LazyVector< RHS_TYPE >` `RHS_LAZY`
- typedef `Rcpp::traits::r_vector_element_converter< RESULT_R_TYPE >::type` `converter_type`
- typedef `Rcpp::traits::storage_type< RESULT_R_TYPE >::type` `STORAGE`

## Public Member Functions

- `Outer` (const `LHS_TYPE` &lhs\_, const `RHS_TYPE` &rhs\_, `Function` fun\_)
- `STORAGE operator()` (int i, int j) const
- `R_xlen_t size` () const
- int `nrow` () const
- int `ncol` () const

## Static Public Attributes

- static const int `RESULT_R_TYPE`

## Private Attributes

- `LHS_LAZY` lhs
- `RHS_LAZY` rhs
- `Function` fun
- int `nr`
- int `nc`

## Additional Inherited Members

### 6.475.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T, typename Function>
class Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >
```

Definition at line 32 of file outer.h.

### 6.475.2 Member Typedef Documentation

### 6.475.2.1 converter\_type

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef Rcpp::traits::r_vector_element_converter<RESULT_R_TYPE>::type Rcpp::sugar::Outer< RTYPE,
LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::converter_type
```

Definition at line 50 of file outer.h.

### 6.475.2.2 LHS\_LAZY

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef Rcpp::internal::LazyVector<LHS_TYPE> Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA,
RHS_T, Function >::LHS_LAZY
```

Definition at line 47 of file outer.h.

### 6.475.2.3 LHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef Rcpp::VectorBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA,
RHS_T, Function >::LHS_TYPE
```

Definition at line 44 of file outer.h.

### 6.475.2.4 result\_type

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef ::Rcpp::traits::result_of<Function>::type Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS←
_NA, RHS_T, Function >::result_type
```

Definition at line 40 of file outer.h.

### 6.475.2.5 RHS\_LAZY

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef Rcpp::internal::LazyVector<RHS_TYPE> Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA,
RHS_T, Function >::RHS_LAZY
```

Definition at line 48 of file outer.h.

### 6.475.2.6 RHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA,
RHS_T, Function >::RHS_TYPE
```

Definition at line 45 of file outer.h.

### 6.475.2.7 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
typedef Rcpp::traits::storage_type<RESULT_R_TYPE>::type Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T,
RHS_NA, RHS_T, Function >::STORAGE
```

Definition at line 51 of file outer.h.

## 6.475.3 Constructor & Destructor Documentation

### 6.475.3.1 Outer()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::Outer (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_,
    Function fun_ ) [inline]
```

Definition at line 53 of file outer.h.

## 6.475.4 Member Function Documentation

### 6.475.4.1 ncol()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
int Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::ncol ( ) const [inline]
```

Definition at line 62 of file outer.h.

References Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::nc.

### 6.475.4.2 nrow()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
int Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::nrow ( ) const [inline]
```

Definition at line 61 of file outer.h.

References Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::nr.

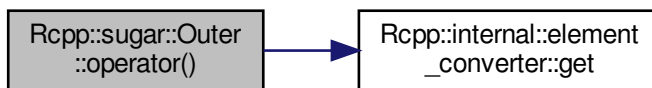
### 6.475.4.3 operator()()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
STORAGE Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::operator() (
    int i,
    int j ) const [inline]
```

Definition at line 56 of file outer.h.

References Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::fun, Rcpp::internal::element\_converter< RTYPE >::get(), Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::lhs, and Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::rhs.

Here is the call graph for this function:



#### 6.475.4.4 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
R_xlen_t Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::size ( ) const
[inline]
```

Definition at line 60 of file outer.h.

References Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::nc, and Rcpp::sugar::←  
Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::nr.

### 6.475.5 Member Data Documentation

#### 6.475.5.1 fun

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
Function Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::fun [private]
```

Definition at line 69 of file outer.h.

Referenced by Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::operator(()).

#### 6.475.5.2 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
LHS_LAZY Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::lhs [private]
```

Definition at line 66 of file outer.h.

Referenced by Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::operator(()).

#### 6.475.5.3 nc

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
int Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::nc [private]
```

Definition at line 70 of file outer.h.

Referenced by Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::ncol(), and Rcpp←  
::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::size().

#### 6.475.5.4 nr

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
int Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::nr [private]
```

Definition at line 70 of file outer.h.

Referenced by Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::nrow(), and Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::size().

#### 6.475.5.5 RESULT\_R\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
const int Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::RESULT_R_TYPE
[static]
```

##### Initial value:

```
=
    Rcpp::traits::r_sexptype_traits<result_type>::rtype
```

Definition at line 41 of file outer.h.

#### 6.475.5.6 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T , typename Function
>
RHS_LAZY Rcpp::sugar::Outer< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T, Function >::rhs [private]
```

Definition at line 67 of file outer.h.

Referenced by Rcpp::sugar::Outer< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T, Function >::operator()().

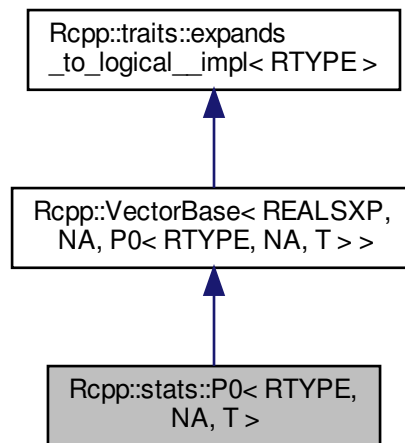
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/outer.h

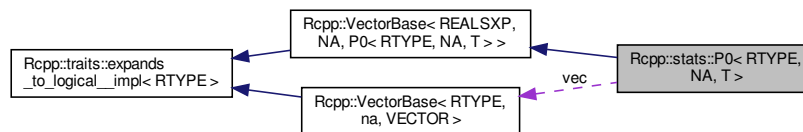
## 6.476 Rcpp::stats::P0< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::P0< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::P0< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `double(* FunPtr)` `(double, int, int)`

### Public Member Functions

- `P0` (`FunPtr ptr_`, `const VEC_TYPE &vec_`, `bool lower_tail=true`, `bool log_=false`)
- `double operator[]` (`int i`) `const`
- `int size` () `const`



## Private Attributes

- [FunPtr ptr](#)
- const [VEC\\_TYPE](#) & *vec*
- int [lower](#)
- int [log](#)

### 6.476.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::stats::P0< RTYPE, NA, T >
```

Definition at line 123 of file dpq.h.

### 6.476.2 Member Typedef Documentation

#### 6.476.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >  
typedef double(* Rcpp::stats::P0< RTYPE, NA, T >::FunPtr) (double, int, int)
```

Definition at line 126 of file dpq.h.

#### 6.476.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::P0< RTYPE, NA, T >::VEC\\_TYPE
```

Definition at line 125 of file dpq.h.

### 6.476.3 Constructor & Destructor Documentation

### 6.476.3.1 P0()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::P0< RTYPE, NA, T >::P0 (
    FunPtr ptr_,
    const VEC_TYPE & vec_,
    bool lower_tail = true,
    bool log_ = false ) [inline]
```

Definition at line 128 of file dpq.h.

## 6.476.4 Member Function Documentation

### 6.476.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P0< RTYPE, NA, T >::operator[] (
    int i ) const [inline]
```

Definition at line 132 of file dpq.h.

References `Rcpp::stats::P0< RTYPE, NA, T >::log`, `Rcpp::stats::P0< RTYPE, NA, T >::lower`, `Rcpp::stats::P0< RTYPE, NA, T >::ptr`, and `Rcpp::stats::P0< RTYPE, NA, T >::vec`.

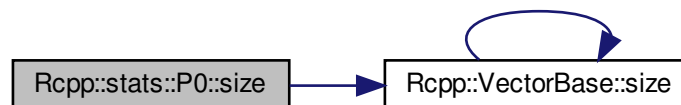
### 6.476.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P0< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 136 of file dpq.h.

References `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`, and `Rcpp::stats::P0< RTYPE, NA, T >::vec`.

Here is the call graph for this function:



## 6.476.5 Member Data Documentation

### 6.476.5.1 log

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::P0< RTYPE, NA, T >::log [private]
```

Definition at line 141 of file dpq.h.

Referenced by Rcpp::stats::P0< RTYPE, NA, T >::operator[]().

### 6.476.5.2 lower

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::P0< RTYPE, NA, T >::lower [private]
```

Definition at line 141 of file dpq.h.

Referenced by Rcpp::stats::P0< RTYPE, NA, T >::operator[]().

### 6.476.5.3 ptr

```
template<int RTYPE, bool NA, typename T >  
FunPtr Rcpp::stats::P0< RTYPE, NA, T >::ptr [private]
```

Definition at line 139 of file dpq.h.

Referenced by Rcpp::stats::P0< RTYPE, NA, T >::operator[]().

### 6.476.5.4 vec

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::stats::P0< RTYPE, NA, T >::vec [private]
```

Definition at line 140 of file dpq.h.

Referenced by Rcpp::stats::P0< RTYPE, NA, T >::operator[](), and Rcpp::stats::P0< RTYPE, NA, T >::size().

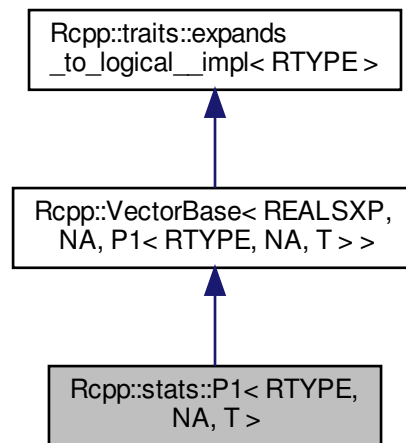
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

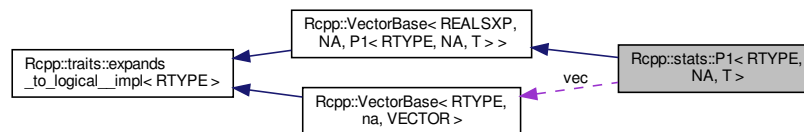
## 6.477 Rcpp::stats::P1< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::P1< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::P1< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `double(* FunPtr) (double, double, int, int)`

### Public Member Functions

- `P1 (FunPtr ptr_, const VEC_TYPE &vec_, double p0_, bool lower_tail=true, bool log_=false)`
- `double operator[] (int i) const`
- `int size () const`

## Private Attributes

- [FunPtr ptr](#)
- const [VEC\\_TYPE](#) & *vec*
- double *p0*
- int *lower*
- int *log*

### 6.477.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::stats::P1< RTYPE, NA, T >
```

Definition at line 147 of file dpq.h.

### 6.477.2 Member Typedef Documentation

#### 6.477.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >  
typedef double(* Rcpp::stats::P1< RTYPE, NA, T >::FunPtr) (double, double, int, int)
```

Definition at line 150 of file dpq.h.

#### 6.477.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::P1< RTYPE, NA, T >::VEC\_TYPE
```

Definition at line 149 of file dpq.h.

### 6.477.3 Constructor & Destructor Documentation

### 6.477.3.1 P1()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::P1< RTYPE, NA, T >::P1 (
    FunPtr ptr_,
    const VEC_TYPE & vec_,
    double p0_,
    bool lower_tail = true,
    bool log_ = false ) [inline]
```

Definition at line 152 of file dpq.h.

## 6.477.4 Member Function Documentation

### 6.477.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P1< RTYPE, NA, T >::operator[] (
    int i ) const [inline]
```

Definition at line 156 of file dpq.h.

References `Rcpp::stats::P1< RTYPE, NA, T >::log`, `Rcpp::stats::P1< RTYPE, NA, T >::lower`, `Rcpp::stats::P1< RTYPE, NA, T >::p0`, `Rcpp::stats::P1< RTYPE, NA, T >::ptr`, and `Rcpp::stats::P1< RTYPE, NA, T >::vec`.

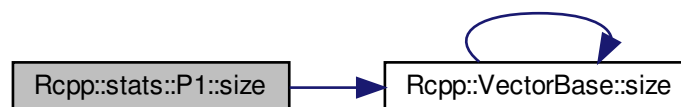
### 6.477.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P1< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 160 of file dpq.h.

References `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`, and `Rcpp::stats::P1< RTYPE, NA, T >::vec`.

Here is the call graph for this function:



## 6.477.5 Member Data Documentation

### 6.477.5.1 log

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::P1< RTYPE, NA, T >::log [private]
```

Definition at line 166 of file dpq.h.

Referenced by Rcpp::stats::P1< RTYPE, NA, T >::operator[]().

### 6.477.5.2 lower

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::P1< RTYPE, NA, T >::lower [private]
```

Definition at line 166 of file dpq.h.

Referenced by Rcpp::stats::P1< RTYPE, NA, T >::operator[]().

### 6.477.5.3 p0

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::P1< RTYPE, NA, T >::p0 [private]
```

Definition at line 165 of file dpq.h.

Referenced by Rcpp::stats::P1< RTYPE, NA, T >::operator[]().

### 6.477.5.4 ptr

```
template<int RTYPE, bool NA, typename T >  
FunPtr Rcpp::stats::P1< RTYPE, NA, T >::ptr [private]
```

Definition at line 163 of file dpq.h.

Referenced by Rcpp::stats::P1< RTYPE, NA, T >::operator[]().

### 6.477.5.5 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::P1< RTYPE, NA, T >::vec [private]
```

Definition at line 164 of file dpq.h.

Referenced by `Rcpp::stats::P1< RTYPE, NA, T >::operator[]()`, and `Rcpp::stats::P1< RTYPE, NA, T >::size()`.

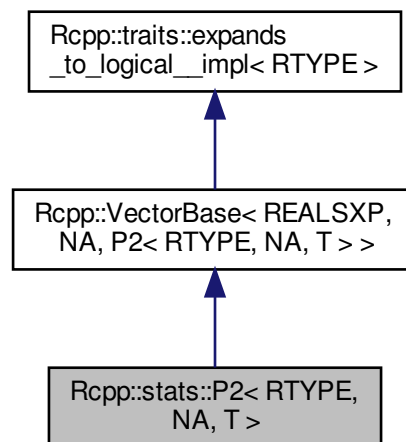
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/stats/dpq/dpq.h`

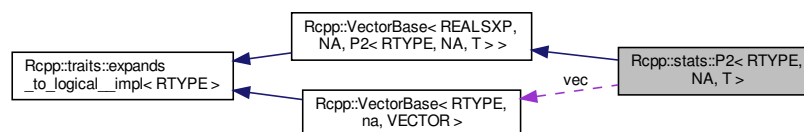
## 6.478 Rcpp::stats::P2< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for `Rcpp::stats::P2< RTYPE, NA, T >`:



Collaboration diagram for `Rcpp::stats::P2< RTYPE, NA, T >`:





## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, NA, T > [VEC\\_TYPE](#)
- typedef double(\* [FunPtr](#)) (double, double, double, int, int)

## Public Member Functions

- [P2](#) ([FunPtr](#) ptr\_, const [VEC\\_TYPE](#) &vec\_, double p0\_, double p1\_, bool lower\_tail=true, bool log\_=false)
- double [operator\[\]](#) (int i) const
- int [size](#) () const

## Private Attributes

- [FunPtr](#) ptr
- const [VEC\\_TYPE](#) & vec
- double p0
- double p1
- int lower
- int log

### 6.478.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::stats::P2< RTYPE, NA, T >
```

Definition at line 172 of file dpq.h.

### 6.478.2 Member Typedef Documentation

#### 6.478.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >  
typedef double(* Rcpp::stats::P2< RTYPE, NA, T >::FunPtr) (double, double, double, int, int)
```

Definition at line 175 of file dpq.h.

### 6.478.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::P2< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 174 of file dpq.h.

## 6.478.3 Constructor & Destructor Documentation

### 6.478.3.1 P2()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::P2< RTYPE, NA, T >::P2 (
    FunPtr ptr_,
    const VEC_TYPE & vec_,
    double p0_,
    double p1_,
    bool lower_tail = true,
    bool log_ = false ) [inline]
```

Definition at line 177 of file dpq.h.

## 6.478.4 Member Function Documentation

### 6.478.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P2< RTYPE, NA, T >::operator[] (
    int i ) const [inline]
```

Definition at line 181 of file dpq.h.

References [Rcpp::stats::P2< RTYPE, NA, T >::log](#), [Rcpp::stats::P2< RTYPE, NA, T >::lower](#), [Rcpp::stats::P2< RTYPE, NA, T >::p0](#), [Rcpp::stats::P2< RTYPE, NA, T >::p1](#), [Rcpp::stats::P2< RTYPE, NA, T >::ptr](#), and [Rcpp::stats::P2< RTYPE, NA, T >::vec](#).

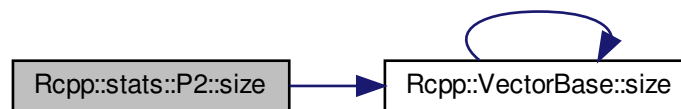
### 6.478.4.2 size()

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::P2< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 185 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::P2< RTYPE, NA, T >::vec.

Here is the call graph for this function:



## 6.478.5 Member Data Documentation

### 6.478.5.1 log

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::P2< RTYPE, NA, T >::log [private]
```

Definition at line 191 of file dpq.h.

Referenced by Rcpp::stats::P2< RTYPE, NA, T >::operator[]().

### 6.478.5.2 lower

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::P2< RTYPE, NA, T >::lower [private]
```

Definition at line 191 of file dpq.h.

Referenced by Rcpp::stats::P2< RTYPE, NA, T >::operator[]().

### 6.478.5.3 p0

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::P2< RTYPE, NA, T >::p0 [private]
```

Definition at line 190 of file dpq.h.

Referenced by Rcpp::stats::P2< RTYPE, NA, T >::operator[ ]().

### 6.478.5.4 p1

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::P2< RTYPE, NA, T >::p1 [private]
```

Definition at line 190 of file dpq.h.

Referenced by Rcpp::stats::P2< RTYPE, NA, T >::operator[ ]().

### 6.478.5.5 ptr

```
template<int RTYPE, bool NA, typename T >  
FunPtr Rcpp::stats::P2< RTYPE, NA, T >::ptr [private]
```

Definition at line 188 of file dpq.h.

Referenced by Rcpp::stats::P2< RTYPE, NA, T >::operator[ ]().

### 6.478.5.6 vec

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::stats::P2< RTYPE, NA, T >::vec [private]
```

Definition at line 189 of file dpq.h.

Referenced by Rcpp::stats::P2< RTYPE, NA, T >::operator[ ](), and Rcpp::stats::P2< RTYPE, NA, T >::size().

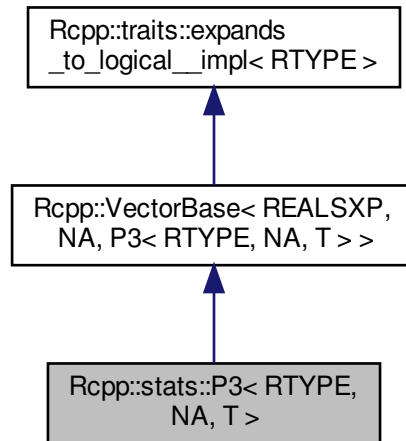
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/dpq/dpq.h](#)

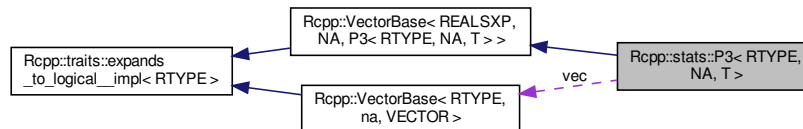
## 6.479 Rcpp::stats::P3< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::P3< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::P3< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `double(* FunPtr)` (`double, double, double, double, int, int`)

### Public Member Functions

- `P3` (`FunPtr ptr_`, `const VEC_TYPE &vec_`, `double p0_`, `double p1_`, `double p2_`, `bool lower_tail=true`, `bool log_↵=false`)
- `double operator[]` (`int i`) `const`
- `int size` () `const`

## Private Attributes

- [FunPtr ptr](#)
- const [VEC\\_TYPE](#) & [vec](#)
- double [p0](#)
- double [p1](#)
- double [p2](#)
- int [lower](#)
- int [log](#)

### 6.479.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::stats::P3< RTYPE, NA, T >
```

Definition at line 195 of file dpq.h.

### 6.479.2 Member Typedef Documentation

#### 6.479.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >  
typedef double(* Rcpp::stats::P3< RTYPE, NA, T >::FunPtr) (double, double, double, double, int,  
int)
```

Definition at line 198 of file dpq.h.

#### 6.479.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::P3< RTYPE, NA, T >::VEC\_TYPE
```

Definition at line 197 of file dpq.h.

### 6.479.3 Constructor & Destructor Documentation

### 6.479.3.1 P3()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::P3< RTYPE, NA, T >::P3 (
    FunPtr ptr_,
    const VEC_TYPE & vec_,
    double p0_,
    double p1_,
    double p2_,
    bool lower_tail = true,
    bool log_ = false ) [inline]
```

Definition at line 200 of file dpq.h.

## 6.479.4 Member Function Documentation

### 6.479.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P3< RTYPE, NA, T >::operator[] (
    int i ) const [inline]
```

Definition at line 204 of file dpq.h.

References Rcpp::stats::P3< RTYPE, NA, T >::log, Rcpp::stats::P3< RTYPE, NA, T >::lower, Rcpp::stats::P3< RTYPE, NA, T >::p0, Rcpp::stats::P3< RTYPE, NA, T >::p1, Rcpp::stats::P3< RTYPE, NA, T >::p2, Rcpp::stats::P3< RTYPE, NA, T >::ptr, and Rcpp::stats::P3< RTYPE, NA, T >::vec.

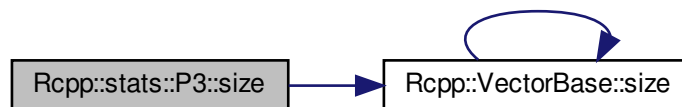
### 6.479.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::P3< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 208 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::P3< RTYPE, NA, T >::vec.

Here is the call graph for this function:



## 6.479.5 Member Data Documentation

### 6.479.5.1 log

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::P3< RTYPE, NA, T >::log [private]
```

Definition at line 214 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[]().

### 6.479.5.2 lower

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::P3< RTYPE, NA, T >::lower [private]
```

Definition at line 214 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[]().

### 6.479.5.3 p0

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::P3< RTYPE, NA, T >::p0 [private]
```

Definition at line 213 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[]().

### 6.479.5.4 p1

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::P3< RTYPE, NA, T >::p1 [private]
```

Definition at line 213 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[]().



### 6.479.5.5 p2

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::P3< RTYPE, NA, T >::p2 [private]
```

Definition at line 213 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[]().

### 6.479.5.6 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::P3< RTYPE, NA, T >::ptr [private]
```

Definition at line 211 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[]().

### 6.479.5.7 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::P3< RTYPE, NA, T >::vec [private]
```

Definition at line 212 of file dpq.h.

Referenced by Rcpp::stats::P3< RTYPE, NA, T >::operator[](), and Rcpp::stats::P3< RTYPE, NA, T >::size().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/dpq/dpq.h](#)

## 6.480 Rcpp::attributes::Param Class Reference

### Public Member Functions

- [Param](#) ()
- [Param](#) (const std::string &paramText)
- bool [empty](#) () const
- bool [operator==](#) (const [Param](#) &other) const
- bool [operator!=](#) (const [Param](#) &other) const
- const std::string & [name](#) () const
- const std::string & [value](#) () const

## Private Attributes

- `std::string` `name_`
- `std::string` `value_`

### 6.480.1 Detailed Description

Definition at line 289 of file `attributes.cpp`.

### 6.480.2 Constructor & Destructor Documentation

#### 6.480.2.1 Param() [1/2]

```
Rcpp::attributes::Param::Param ( ) [inline]
```

Definition at line 291 of file `attributes.cpp`.

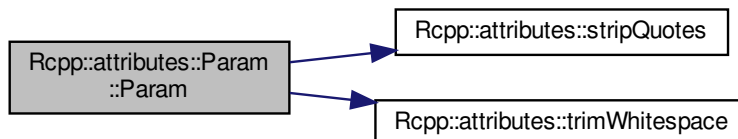
#### 6.480.2.2 Param() [2/2]

```
Rcpp::attributes::Param::Param (
    const std::string & paramText ) [explicit]
```

Definition at line 1061 of file `attributes.cpp`.

References `name_`, `Rcpp::attributes::stripQuotes()`, `Rcpp::attributes::trimWhitespace()`, and `value_`.

Here is the call graph for this function:



## 6.480.3 Member Function Documentation

### 6.480.3.1 empty()

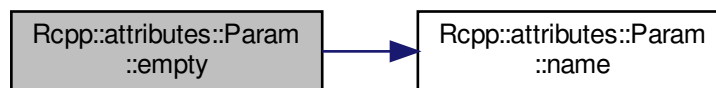
```
bool Rcpp::attributes::Param::empty ( ) const [inline]
```

Definition at line 293 of file attributes.cpp.

References `name()`.

Referenced by `Rcpp::attributes::Attribute::hasParameter()`, `Rcpp::attributes::Attribute::invisible()`, `Rcpp::attributes::operator<<()`, and `Rcpp::attributes::Attribute::rng()`.

Here is the call graph for this function:



### 6.480.3.2 name()

```
const std::string& Rcpp::attributes::Param::name ( ) const [inline]
```

Definition at line 305 of file attributes.cpp.

References `name_`.

Referenced by `empty()`, and `Rcpp::attributes::operator<<()`.

### 6.480.3.3 operator"!="()

```
bool Rcpp::attributes::Param::operator!= (
    const Param & other ) const [inline]
```

Definition at line 300 of file attributes.cpp.

#### 6.480.3.4 operator==( )

```
bool Rcpp::attributes::Param::operator== (
    const Param & other ) const [inline]
```

Definition at line 295 of file attributes.cpp.

References `name_`, and `value_`.

#### 6.480.3.5 value( )

```
const std::string& Rcpp::attributes::Param::value ( ) const [inline]
```

Definition at line 306 of file attributes.cpp.

References `value_`.

Referenced by `Rcpp::attributes::Attribute::customRSignature()`, `Rcpp::attributes::Attribute::exportedName()`, `Rcpp::attributes::Attribute::invisible()`, `Rcpp::attributes::operator<<()`, and `Rcpp::attributes::Attribute::rng()`.

### 6.480.4 Member Data Documentation

#### 6.480.4.1 name\_

```
std::string Rcpp::attributes::Param::name_ [private]
```

Definition at line 309 of file attributes.cpp.

Referenced by `name()`, `operator==( )`, and `Param()`.

#### 6.480.4.2 value\_

```
std::string Rcpp::attributes::Param::value_ [private]
```

Definition at line 310 of file attributes.cpp.

Referenced by `operator==( )`, `Param()`, and `value()`.

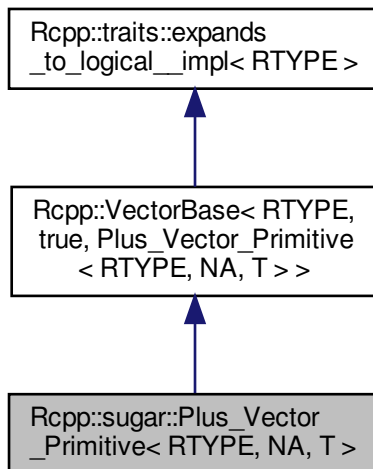
The documentation for this class was generated from the following file:

- [src/attributes.cpp](#)

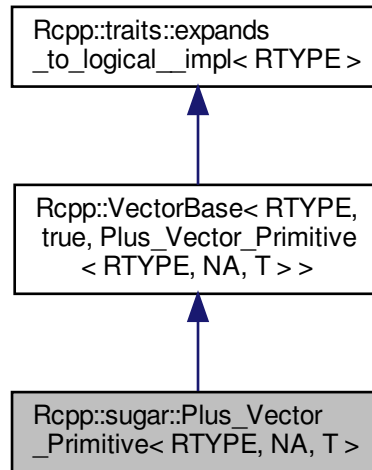
## 6.481 Rcpp::sugar::Plus\_Vector\_Primitive< RTYPE, NA, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Primitive< RTYPE, NA, T >:



Collaboration diagram for `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >`:



## Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::traits::Extractor< RTYPE, NA, T >::type` `EXT`

## Public Member Functions

- `Plus_Vector_Primitive` (const `VEC_TYPE` &lhs\_, `STORAGE` rhs\_)
- `STORAGE operator[]` (`R_xlen_t` i) const
- `R_xlen_t size` () const

## Private Attributes

- const `EXT` &lhs
- `STORAGE` rhs
- bool rhs\_na

### 6.481.1 Detailed Description

```

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >
  
```

Definition at line 248 of file plus.h.

## 6.481.2 Member Typedef Documentation

### 6.481.2.1 EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Plus_Vector_Primitive< RTYPE,
NA, T >::EXT
```

Definition at line 254 of file plus.h.

### 6.481.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >↔
::STORAGE
```

Definition at line 252 of file plus.h.

### 6.481.2.3 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 251 of file plus.h.

## 6.481.3 Constructor & Destructor Documentation

### 6.481.3.1 Plus\_Vector\_Primitive()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::Plus_Vector_Primitive (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 256 of file plus.h.

## 6.481.4 Member Function Documentation

### 6.481.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 260 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs`, `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs`, and `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs_na`.

### 6.481.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 266 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs`.

## 6.481.5 Member Data Documentation

### 6.481.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs [private]
```

Definition at line 269 of file plus.h.

Referenced by `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::operator[]()`, `Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::operator[]()`, `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::operator[]()`, `Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::operator[]()`, `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::size()`, `Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::size()`, `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::size()`, and `Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::size()`.



### 6.481.5.2 rhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs [private]
```

Definition at line 270 of file plus.h.

Referenced by Rcpp::sugar::Plus\_Vector\_Primitive< RTYPE, NA, T >::operator[](), Rcpp::sugar::Plus\_Vector\_Primitive< REALSXP, NA, T >::operator[](), Rcpp::sugar::Plus\_Vector\_Primitive< RTYPE, false, T >::operator[](), and Rcpp::sugar::Plus\_Vector\_Primitive< REALSXP, false, T >::operator[]().

### 6.481.5.3 rhs\_na

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs_na [private]
```

Definition at line 271 of file plus.h.

Referenced by Rcpp::sugar::Plus\_Vector\_Primitive< RTYPE, NA, T >::operator[](), and Rcpp::sugar::Plus\_Vector\_Primitive< RTYPE, false, T >::operator[]().

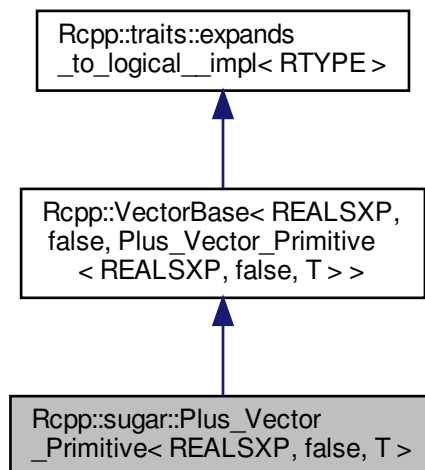
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/plus.h](#)

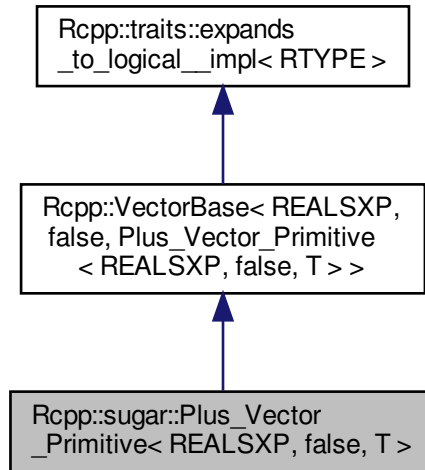
## 6.482 Rcpp::sugar::Plus\_Vector\_Primitive< REALSXP, false, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Primitive< REALSXP, false, T >:



Collaboration diagram for `Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >`:



## Public Types

- typedef `Rcpp::VectorBase< REALSXP, false, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< REALSXP, false, T >::type` `EXT`

## Public Member Functions

- `Plus_Vector_Primitive` (const `VEC_TYPE` &lhs\_, double rhs\_)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const `EXT` & lhs
- double rhs

### 6.482.1 Detailed Description

```

template<typename T>
class Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >
  
```

Definition at line 323 of file plus.h.

## 6.482.2 Member Typedef Documentation

### 6.482.2.1 EXT

```
template<typename T >
typedef Rcpp::traits::Extractor< REALSXP, false, T>::type Rcpp::sugar::Plus_Vector_Primitive<
REALSXP, false, T >::EXT
```

Definition at line 328 of file plus.h.

### 6.482.2.2 VEC\_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP, false, T> Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T
>::VEC_TYPE
```

Definition at line 326 of file plus.h.

## 6.482.3 Constructor & Destructor Documentation

### 6.482.3.1 Plus\_Vector\_Primitive()

```
template<typename T >
Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::Plus_Vector_Primitive (
    const VEC_TYPE & lhs_,
    double rhs_ ) [inline]
```

Definition at line 330 of file plus.h.

## 6.482.4 Member Function Documentation

#### 6.482.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::operator[] (
    R_xlen_t i) const [inline]
```

Definition at line 333 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs`, and `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs`.

#### 6.482.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::size ( ) const [inline]
```

Definition at line 337 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs`.

### 6.482.5 Member Data Documentation

#### 6.482.5.1 lhs

```
template<typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::lhs [private]
```

Definition at line 340 of file plus.h.

#### 6.482.5.2 rhs

```
template<typename T >
double Rcpp::sugar::Plus_Vector_Primitive< REALSXP, false, T >::rhs [private]
```

Definition at line 341 of file plus.h.

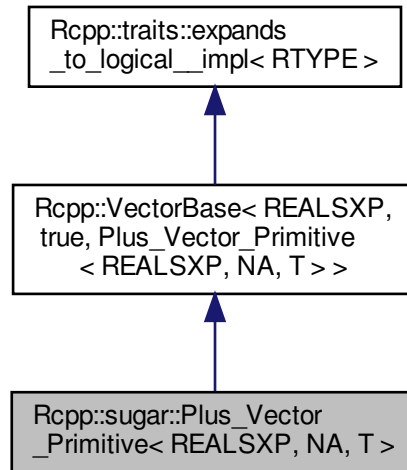
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/plus.h](#)

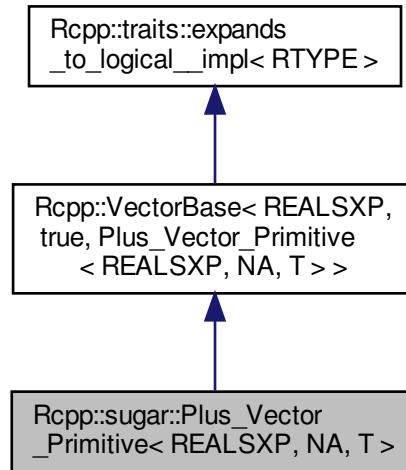
## 6.483 Rcpp::sugar::Plus\_Vector\_Primitive< REALSXP, NA, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Primitive< REALSXP, NA, T >:



Collaboration diagram for `Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >`:



## Public Types

- typedef `Rcpp::VectorBase< REALSXP, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< REALSXP, NA, T >::type` `EXT`

## Public Member Functions

- `Plus_Vector_Primitive` (const `VEC_TYPE` &lhs\_, double rhs\_)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const `EXT` & lhs
- double rhs

### 6.483.1 Detailed Description

```

template<bool NA, typename T>
class Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >
  
```

Definition at line 276 of file plus.h.

## 6.483.2 Member Typedef Documentation

### 6.483.2.1 EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor< REALSXP, NA, T>::type Rcpp::sugar::Plus_Vector_Primitive< REALSXP,
NA, T >::EXT
```

Definition at line 280 of file plus.h.

### 6.483.2.2 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >↔
::VEC_TYPE
```

Definition at line 279 of file plus.h.

## 6.483.3 Constructor & Destructor Documentation

### 6.483.3.1 Plus\_Vector\_Primitive()

```
template<bool NA, typename T >
Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::Plus_Vector_Primitive (
    const VEC_TYPE & lhs_,
    double rhs_ ) [inline]
```

Definition at line 282 of file plus.h.

## 6.483.4 Member Function Documentation

#### 6.483.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::operator[] (
    R_xlen_t i) const [inline]
```

Definition at line 286 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs`, and `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs`.

#### 6.483.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::size ( ) const [inline]
```

Definition at line 290 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs`.

### 6.483.5 Member Data Documentation

#### 6.483.5.1 lhs

```
template<bool NA, typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::lhs [private]
```

Definition at line 293 of file plus.h.

#### 6.483.5.2 rhs

```
template<bool NA, typename T >
double Rcpp::sugar::Plus_Vector_Primitive< REALSXP, NA, T >::rhs [private]
```

Definition at line 294 of file plus.h.

The documentation for this class was generated from the following file:

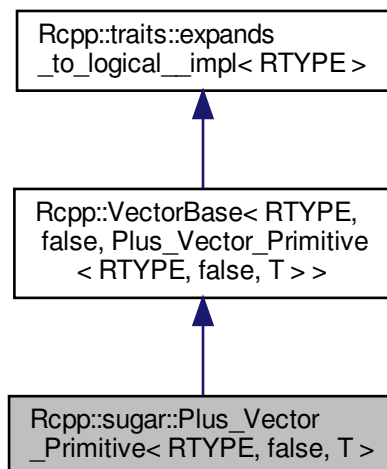
- [inst/include/Rcpp/sugar/operators/plus.h](#)



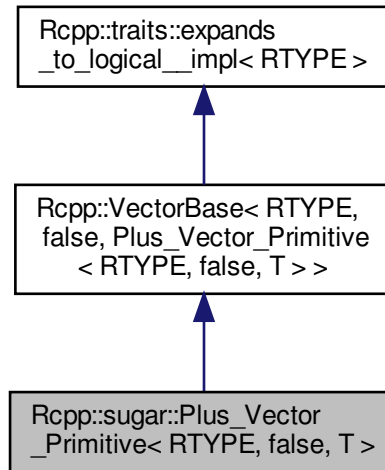
## 6.484 Rcpp::sugar::Plus\_Vector\_Primitive< RTYPE, false, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Primitive< RTYPE, false, T >:



Collaboration diagram for `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >`:



## Public Types

- typedef `Rcpp::VectorBase< RTYPE, false, T >` `VEC_TYPE`
- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::traits::Extractor< RTYPE, false, T >::type` `EXT`

## Public Member Functions

- `Plus_Vector_Primitive` (const `VEC_TYPE` &lhs\_, `STORAGE` rhs\_)
- `STORAGE operator[]` (R\_xlen\_t i) const
- `R_xlen_t size` () const

## Private Attributes

- const `EXT` & lhs
- `STORAGE` rhs
- bool rhs\_na

### 6.484.1 Detailed Description

```

template<int RTYPE, typename T>
class Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >
  
```

Definition at line 300 of file plus.h.

## 6.484.2 Member Typedef Documentation

### 6.484.2.1 EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor< RTYPE, false, T>::type Rcpp::sugar::Plus_Vector_Primitive< RTYPE,
false, T >::EXT
```

Definition at line 305 of file plus.h.

### 6.484.2.2 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T
>::STORAGE
```

Definition at line 303 of file plus.h.

### 6.484.2.3 VEC\_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >↔
::VEC_TYPE
```

Definition at line 302 of file plus.h.

## 6.484.3 Constructor & Destructor Documentation

### 6.484.3.1 Plus\_Vector\_Primitive()

```
template<int RTYPE, typename T >
Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::Plus_Vector_Primitive (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 307 of file plus.h.

## 6.484.4 Member Function Documentation

### 6.484.4.1 operator[]()

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::operator[] (   
    R_xlen_t i ) const [inline]
```

Definition at line 310 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs`, `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs`, and `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::rhs_na`.

### 6.484.4.2 size()

```
template<int RTYPE, typename T >  
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::size ( ) const [inline]
```

Definition at line 314 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Primitive< RTYPE, NA, T >::lhs`.

## 6.484.5 Member Data Documentation

### 6.484.5.1 lhs

```
template<int RTYPE, typename T >  
const EXT& Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::lhs [private]
```

Definition at line 317 of file plus.h.

### 6.484.5.2 rhs

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::rhs [private]
```

Definition at line 318 of file plus.h.

### 6.484.5.3 rhs\_na

```
template<int RTYPE, typename T >  
bool Rcpp::sugar::Plus_Vector_Primitive< RTYPE, false, T >::rhs_na [private]
```

Definition at line 319 of file plus.h.

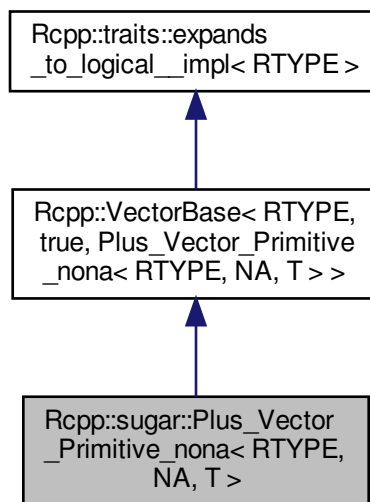
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/plus.h](#)

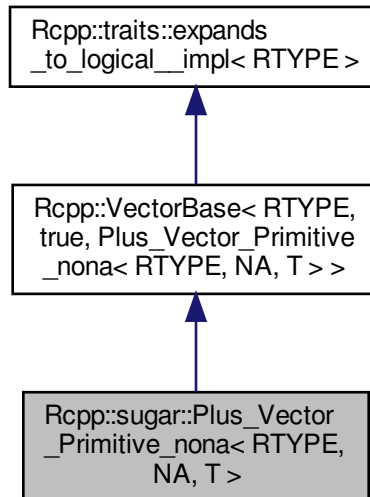
## 6.485 Rcpp::sugar::Plus\_Vector\_Primitive\_nona< RTYPE, NA, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Primitive\_nona< RTYPE, NA, T >:



Collaboration diagram for `Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >`:



## Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::traits::Extractor< RTYPE, NA, T >::type` `EXT`

## Public Member Functions

- `Plus_Vector_Primitive_nona` (const `VEC_TYPE` &lhs\_, `STORAGE` rhs\_)
- `STORAGE operator[]` (`R_xlen_t` i) const
- `R_xlen_t size` () const

## Private Attributes

- const `EXT` & lhs
- `STORAGE` rhs

### 6.485.1 Detailed Description

```

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >
  
```

Definition at line 351 of file plus.h.

## 6.485.2 Member Typedef Documentation

### 6.485.2.1 EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Plus_Vector_Primitive_nona<
RTYPE, NA, T >::EXT
```

Definition at line 355 of file plus.h.

### 6.485.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T
>::STORAGE
```

Definition at line 354 of file plus.h.

### 6.485.2.3 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >↔
::VEC_TYPE
```

Definition at line 353 of file plus.h.

## 6.485.3 Constructor & Destructor Documentation

### 6.485.3.1 Plus\_Vector\_Primitive\_nona()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::Plus_Vector_Primitive_nona (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 357 of file plus.h.

## 6.485.4 Member Function Documentation

### 6.485.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 361 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs`, and `Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::rhs`.

### 6.485.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 366 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs`.

## 6.485.5 Member Data Documentation

### 6.485.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs [private]
```

Definition at line 369 of file plus.h.

Referenced by `Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::operator[]()`, `Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::operator[]()`, `Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::operator[]()`, `Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::operator[]()`, `Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::size()`, `Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::size()`, `Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::size()`, and `Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::size()`.



## 6.485.5.2 rhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::rhs [private]
```

Definition at line 370 of file plus.h.

Referenced by Rcpp::sugar::Plus\_Vector\_Primitive\_nona< RTYPE, NA, T >::operator[](), Rcpp::sugar::Plus\_Vector\_Primitive\_nona< REALSXP, NA, T >::operator[](), Rcpp::sugar::Plus\_Vector\_Primitive\_nona< RTYPE, false, T >::operator[](), and Rcpp::sugar::Plus\_Vector\_Primitive\_nona< REALSXP, false, T >::operator[]().

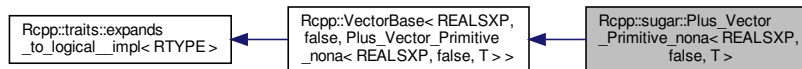
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

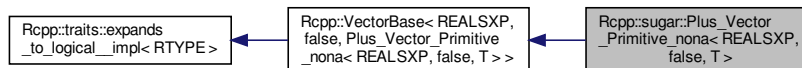
## 6.486 Rcpp::sugar::Plus\_Vector\_Primitive\_nona< REALSXP, false, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Primitive\_nona< REALSXP, false, T >:



Collaboration diagram for Rcpp::sugar::Plus\_Vector\_Primitive\_nona< REALSXP, false, T >:



### Public Types

- typedef [Rcpp::VectorBase< REALSXP, false, T >](#) **VEC\_TYPE**
- typedef [Rcpp::traits::Extractor< REALSXP, false, T >::type](#) **EXT**

## Public Member Functions

- [Plus\\_Vector\\_Primitive\\_nona](#) (const [VEC\\_TYPE](#) &lhs\_, double rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [EXT](#) & lhs
- double [rhs](#)

### 6.486.1 Detailed Description

```
template<typename T>
class Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >
```

Definition at line 422 of file plus.h.

### 6.486.2 Member Typedef Documentation

#### 6.486.2.1 EXT

```
template<typename T >
typedef Rcpp::traits::Extractor< REALSXP, false, T>::type Rcpp::sugar::Plus_Vector_Primitive_nona<
REALSXP, false, T >::EXT
```

Definition at line 426 of file plus.h.

#### 6.486.2.2 VEC\_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP, false, T> Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false,
T >::VEC_TYPE
```

Definition at line 425 of file plus.h.

### 6.486.3 Constructor & Destructor Documentation

### 6.486.3.1 Plus\_Vector\_Primitive\_nona()

```
template<typename T >
Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::Plus_Vector_Primitive_nona (
    const VEC_TYPE & lhs_,
    double rhs_ ) [inline]
```

Definition at line 428 of file plus.h.

## 6.486.4 Member Function Documentation

### 6.486.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 431 of file plus.h.

References [Rcpp::sugar::Plus\\_Vector\\_Primitive\\_nona< RTYPE, NA, T >::lhs](#), and [Rcpp::sugar::Plus\\_Vector\\_Primitive\\_nona< RTYPE, NA, T >::rhs](#).

### 6.486.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::size ( ) const [inline]
```

Definition at line 435 of file plus.h.

References [Rcpp::sugar::Plus\\_Vector\\_Primitive\\_nona< RTYPE, NA, T >::lhs](#).

## 6.486.5 Member Data Documentation

### 6.486.5.1 lhs

```
template<typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::lhs [private]
```

Definition at line 438 of file plus.h.

### 6.486.5.2 rhs

```
template<typename T >
double Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, false, T >::rhs [private]
```

Definition at line 439 of file plus.h.

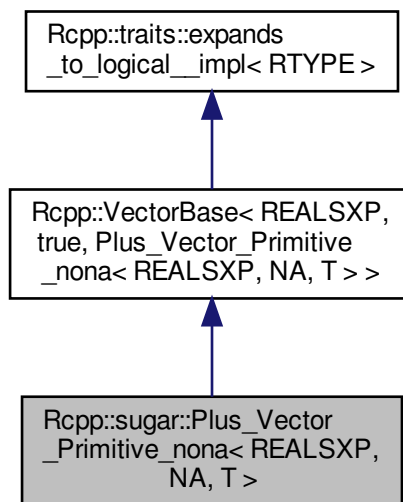
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/plus.h](#)

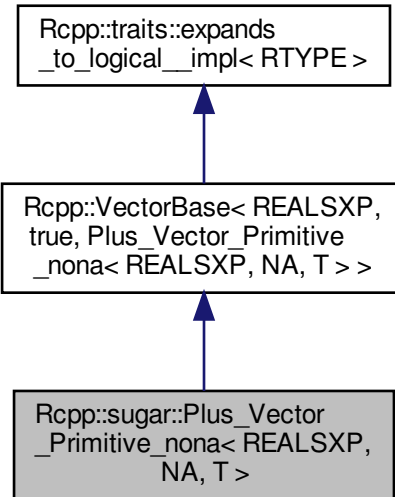
## 6.487 Rcpp::sugar::Plus\_Vector\_Primitive\_nona< REALSXP, NA, T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Primitive\_nona< REALSXP, NA, T >:



Collaboration diagram for Rcpp::sugar::Plus\_Vector\_Primitive\_nona< REALSXP, NA, T >:



## Public Types

- typedef `Rcpp::VectorBase< REALSXP, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< REALSXP, NA, T >::type` `EXT`

## Public Member Functions

- `Plus_Vector_Primitive_nona` (const `VEC_TYPE` &lhs\_, double rhs\_)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const `EXT` & lhs
- double rhs

### 6.487.1 Detailed Description

```

template<bool NA, typename T>
class Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >
  
```

Definition at line 374 of file plus.h.

## 6.487.2 Member Typedef Documentation

### 6.487.2.1 EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP, NA, T>::type Rcpp::sugar::Plus_Vector_Primitive_nona<
REALSXP, NA, T >::EXT
```

Definition at line 378 of file plus.h.

### 6.487.2.2 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T
>::VEC_TYPE
```

Definition at line 377 of file plus.h.

## 6.487.3 Constructor & Destructor Documentation

### 6.487.3.1 Plus\_Vector\_Primitive\_nona()

```
template<bool NA, typename T >
Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::Plus_Vector_Primitive_nona (
    const VEC_TYPE & lhs_,
    double rhs_ ) [inline]
```

Definition at line 380 of file plus.h.

## 6.487.4 Member Function Documentation

### 6.487.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::operator[] (
    R_xlen_t i) const [inline]
```

Definition at line 384 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs`, and `Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::rhs`.

### 6.487.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::size ( ) const [inline]
```

Definition at line 388 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T >::lhs`.

## 6.487.5 Member Data Documentation

### 6.487.5.1 lhs

```
template<bool NA, typename T >
const EXT& Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::lhs [private]
```

Definition at line 391 of file plus.h.

### 6.487.5.2 rhs

```
template<bool NA, typename T >
double Rcpp::sugar::Plus_Vector_Primitive_nona< REALSXP, NA, T >::rhs [private]
```

Definition at line 392 of file plus.h.

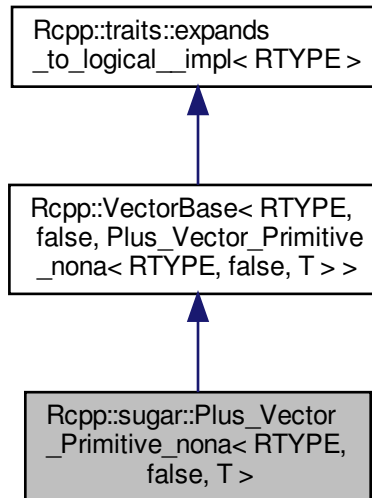
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/operators/plus.h`

## 6.488 Rcpp::sugar::Plus\_Vector\_Primitive\_nona< RTYPE, false, T > Class Template Reference

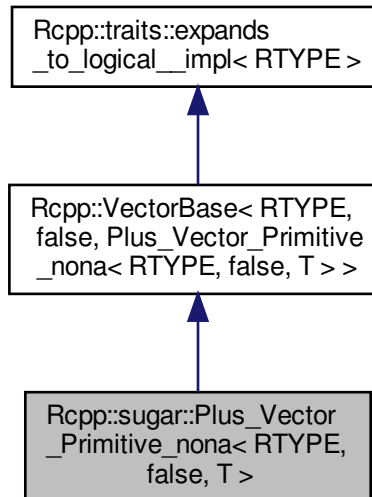
```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Primitive\_nona< RTYPE, false, T >:





Collaboration diagram for Rcpp::sugar::Plus\_Vector\_Primitive\_nona< RTYPE, false, T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, false, T > [VEC\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, T >::type [EXT](#)

## Public Member Functions

- [Plus\\_Vector\\_Primitive\\_nona](#) (const [VEC\\_TYPE](#) &lhs\_, [STORAGE](#) rhs\_)
- [STORAGE](#) operator[] (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [EXT](#) &lhs
- [STORAGE](#) rhs

## 6.488.1 Detailed Description

```

template<int RTYPE, typename T>
class Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >

```

Definition at line 399 of file plus.h.

## 6.488.2 Member Typedef Documentation

### 6.488.2.1 EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor< RTYPE, false, T>::type Rcpp::sugar::Plus_Vector_Primitive_nona<
RTYPE, false, T >::EXT
```

Definition at line 404 of file plus.h.

### 6.488.2.2 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false,
T >::STORAGE
```

Definition at line 402 of file plus.h.

### 6.488.2.3 VEC\_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T
>::VEC_TYPE
```

Definition at line 401 of file plus.h.

## 6.488.3 Constructor & Destructor Documentation

### 6.488.3.1 Plus\_Vector\_Primitive\_nona()

```
template<int RTYPE, typename T >
Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::Plus_Vector_Primitive_nona (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 406 of file plus.h.

## 6.488.4 Member Function Documentation

### 6.488.4.1 operator[]()

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::operator[] (   
    R_xlen_t i ) const [inline]
```

Definition at line 409 of file plus.h.

References [Rcpp::sugar::Plus\\_Vector\\_Primitive\\_nona< RTYPE, NA, T >::lhs](#), and [Rcpp::sugar::Plus\\_Vector\\_Primitive\\_nona< RTYPE, NA, T >::rhs](#).

### 6.488.4.2 size()

```
template<int RTYPE, typename T >  
R_xlen_t Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::size ( ) const [inline]
```

Definition at line 413 of file plus.h.

References [Rcpp::sugar::Plus\\_Vector\\_Primitive\\_nona< RTYPE, NA, T >::lhs](#).

## 6.488.5 Member Data Documentation

### 6.488.5.1 lhs

```
template<int RTYPE, typename T >  
const EXT& Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::lhs [private]
```

Definition at line 416 of file plus.h.

### 6.488.5.2 rhs

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Plus_Vector_Primitive_nona< RTYPE, false, T >::rhs [private]
```

Definition at line 417 of file plus.h.

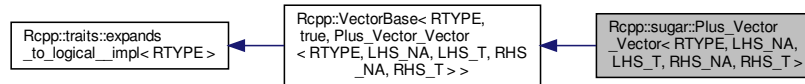
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/plus.h](#)

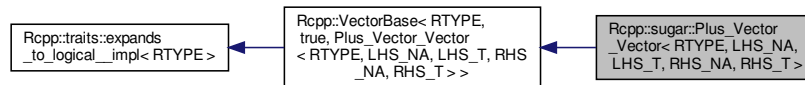
## 6.489 Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, RHS\_NA, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Plus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) & lhs
- const [RHS\\_EXT](#) & rhs

## 6.489.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 29 of file plus.h.

## 6.489.2 Member Typedef Documentation

### 6.489.2.1 LHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 35 of file plus.h.

### 6.489.2.2 LHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE, LHS_NA, LHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 32 of file plus.h.

### 6.489.2.3 RHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 36 of file plus.h.

### 6.489.2.4 RHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 33 of file plus.h.

### 6.489.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 31 of file plus.h.

## 6.489.3 Constructor & Destructor Documentation

### 6.489.3.1 Plus\_Vector\_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Plus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 38 of file plus.h.

## 6.489.4 Member Function Documentation

### 6.489.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 41 of file plus.h.

References Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.489.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 48 of file plus.h.

References Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs.

## 6.489.5 Member Data Documentation

### 6.489.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 51 of file plus.h.

Referenced by Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >::size(), Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >::size(), Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >::size(), and Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >::size().

### 6.489.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs [private]
```

Definition at line 52 of file plus.h.

Referenced by Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >::operator[ ](), Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >::operator[ ](), and Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >::operator[ ]().

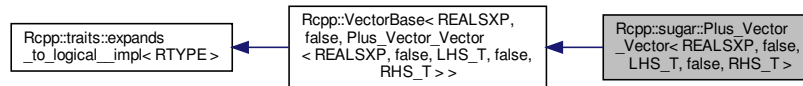
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/plus.h](#)

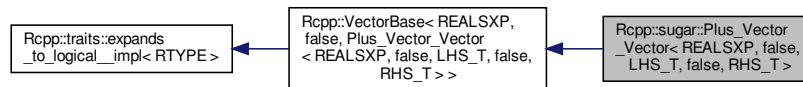
## 6.490 Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase< REALSXP, false, LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase< REALSXP, false, RHS\\_T >](#) [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor< REALSXP, false, LHS\\_T >::type](#) [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor< REALSXP, false, RHS\\_T >::type](#) [RHS\\_EXT](#)

### Public Member Functions

- [Plus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs



## 6.490.1 Detailed Description

```
template<typename LHS_T, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
```

Definition at line 218 of file plus.h.

## 6.490.2 Member Typedef Documentation

### 6.490.2.1 LHS\_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 224 of file plus.h.

### 6.490.2.2 LHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, false, LHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 221 of file plus.h.

### 6.490.2.3 RHS\_EXT

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 225 of file plus.h.

### 6.490.2.4 RHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, false, RHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 222 of file plus.h.

## 6.490.3 Constructor & Destructor Documentation

### 6.490.3.1 Plus\_Vector\_Vector()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::Plus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 227 of file plus.h.

## 6.490.4 Member Function Documentation

### 6.490.4.1 operator[]()

```
template<typename LHS_T , typename RHS_T >
double Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 230 of file plus.h.

References [Rcpp::sugar::Plus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#), and [Rcpp::sugar::Plus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::rhs](#).

### 6.490.4.2 size()

```
template<typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 234 of file plus.h.

References [Rcpp::sugar::Plus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#).

## 6.490.5 Member Data Documentation

### 6.490.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::lhs [private]
```

Definition at line 237 of file plus.h.

### 6.490.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::rhs [private]
```

Definition at line 238 of file plus.h.

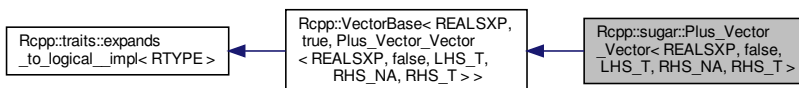
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

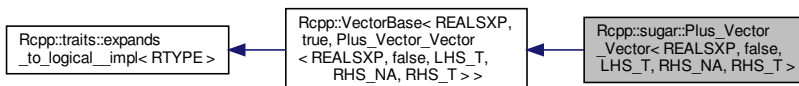
## 6.491 Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< REALSXP, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, RHS\_NA, RHS\_T >::type [RHS\\_EXT](#)

## Public Member Functions

- [Plus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & [lhs](#)
- const [RHS\\_EXT](#) & [rhs](#)

### 6.491.1 Detailed Description

```
template<typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
```

Definition at line 113 of file plus.h.

### 6.491.2 Member Typedef Documentation

#### 6.491.2.1 LHS\_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Plus\_Vector\_Vector<
REALSXP, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 119 of file plus.h.

### 6.491.2.2 LHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, false, LHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 116 of file plus.h.

### 6.491.2.3 RHS\_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 120 of file plus.h.

### 6.491.2.4 RHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, RHS_NA, RHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 117 of file plus.h.

## 6.491.3 Constructor & Destructor Documentation

### 6.491.3.1 Plus\_Vector\_Vector()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::Plus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 122 of file plus.h.

### 6.491.4 Member Function Documentation

### 6.491.4.1 operator[]()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 125 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

### 6.491.4.2 size()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 129 of file plus.h.

References `Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

## 6.491.5 Member Data Documentation

### 6.491.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 132 of file plus.h.

### 6.491.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::rhs [private]
```

Definition at line 133 of file plus.h.

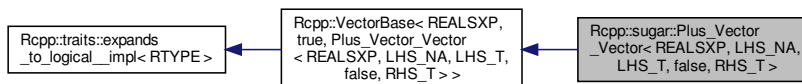
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/plus.h](#)

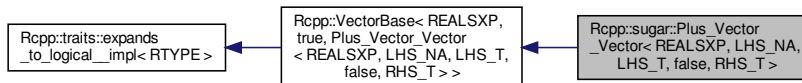
## 6.492 Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< REALSXP, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Plus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

### 6.492.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
```

Definition at line 166 of file plus.h.

### 6.492.2 Member Typedef Documentation

#### 6.492.2.1 LHS\_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 172 of file plus.h.

#### 6.492.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, LHS_NA, LHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 169 of file plus.h.

#### 6.492.2.3 RHS\_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 173 of file plus.h.



#### 6.492.2.4 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, false, RHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 170 of file plus.h.

### 6.492.3 Constructor & Destructor Documentation

#### 6.492.3.1 Plus\_Vector\_Vector()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::Plus_Vector_Vector (
    const RHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 175 of file plus.h.

### 6.492.4 Member Function Documentation

#### 6.492.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
double Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 178 of file plus.h.

References Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

#### 6.492.4.2 size()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 182 of file plus.h.

References Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs.

## 6.492.5 Member Data Documentation

### 6.492.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::lhs [private]
```

Definition at line 185 of file plus.h.

### 6.492.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::rhs [private]
```

Definition at line 186 of file plus.h.

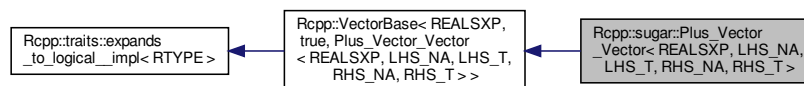
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/plus.h](#)

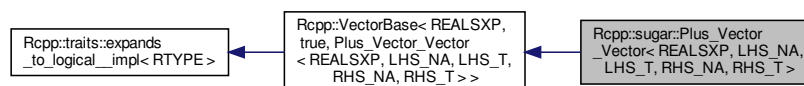
## 6.493 Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Plus\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< REALSXP, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, RHS\_NA, RHS\_T >::type [RHS\\_EXT](#)

## Public Member Functions

- [Plus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & lhs
- const [RHS\\_EXT](#) & rhs

### 6.493.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 59 of file plus.h.

### 6.493.2 Member Typedef Documentation

#### 6.493.2.1 LHS\_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Plus\_Vector\_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 65 of file plus.h.

### 6.493.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, LHS_NA, LHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 62 of file plus.h.

### 6.493.2.3 RHS\_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 66 of file plus.h.

### 6.493.2.4 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, RHS_NA, RHS_T> Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 63 of file plus.h.

## 6.493.3 Constructor & Destructor Documentation

### 6.493.3.1 Plus\_Vector\_Vector()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::Plus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 68 of file plus.h.

### 6.493.4 Member Function Documentation

#### 6.493.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 71 of file plus.h.

References Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

#### 6.493.4.2 size()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 75 of file plus.h.

References Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs.

### 6.493.5 Member Data Documentation

#### 6.493.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 78 of file plus.h.

#### 6.493.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 79 of file plus.h.

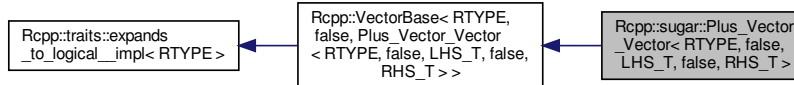
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

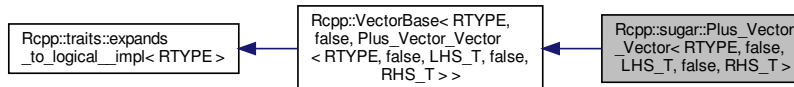
## 6.494 Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Plus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

## 6.494.1 Detailed Description

```
template<int RTYPE, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
```

Definition at line 194 of file plus.h.

## 6.494.2 Member Typedef Documentation

### 6.494.2.1 LHS\_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 200 of file plus.h.

### 6.494.2.2 LHS\_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T,
false, RHS_T >::LHS_TYPE
```

Definition at line 197 of file plus.h.

### 6.494.2.3 RHS\_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 201 of file plus.h.

#### 6.494.2.4 RHS\_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T,
false, RHS_T >::RHS_TYPE
```

Definition at line 198 of file plus.h.

#### 6.494.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T,
false, RHS_T >::STORAGE
```

Definition at line 196 of file plus.h.

### 6.494.3 Constructor & Destructor Documentation

#### 6.494.3.1 Plus\_Vector\_Vector()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::Plus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 203 of file plus.h.

### 6.494.4 Member Function Documentation

#### 6.494.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 206 of file plus.h.

References Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.



### 6.494.4.2 size()

```
template<int RTYPE, typename LHS_T , typename RHS_T >  
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size ( ) const  
[inline]
```

Definition at line 210 of file plus.h.

References [Rcpp::sugar::Plus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#).

## 6.494.5 Member Data Documentation

### 6.494.5.1 lhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >  
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::lhs [private]
```

Definition at line 213 of file plus.h.

### 6.494.5.2 rhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >  
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::rhs [private]
```

Definition at line 214 of file plus.h.

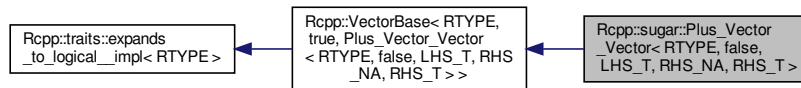
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/plus.h](#)

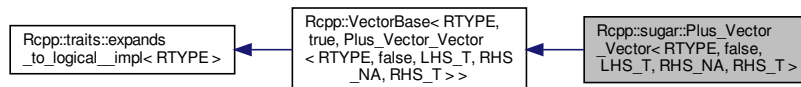
## 6.495 Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, RHS\_NA, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Plus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

## 6.495.1 Detailed Description

```
template<int RTYPE, typename LHS_T, bool RHS_NA, typename RHS_T>  
class Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
```

Definition at line 87 of file plus.h.

## 6.495.2 Member Typedef Documentation

### 6.495.2.1 LHS\_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<  
RTYPE, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 93 of file plus.h.

### 6.495.2.2 LHS\_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T,  
RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 90 of file plus.h.

### 6.495.2.3 RHS\_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<  
RTYPE, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 94 of file plus.h.

### 6.495.2.4 RHS\_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T,
RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 91 of file plus.h.

### 6.495.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 89 of file plus.h.

## 6.495.3 Constructor & Destructor Documentation

### 6.495.3.1 Plus\_Vector\_Vector()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::Plus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 96 of file plus.h.

## 6.495.4 Member Function Documentation

### 6.495.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 99 of file plus.h.

References Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.495.4.2 size()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >  
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size ( ) const  
[inline]
```

Definition at line 105 of file plus.h.

References [Rcpp::sugar::Plus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#).

## 6.495.5 Member Data Documentation

### 6.495.5.1 lhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >  
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 108 of file plus.h.

### 6.495.5.2 rhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >  
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::rhs [private]
```

Definition at line 109 of file plus.h.

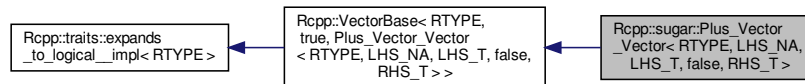
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/plus.h](#)

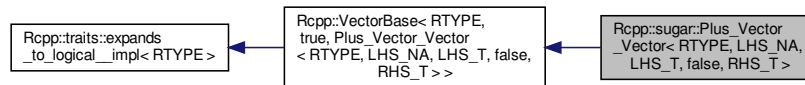
## 6.496 Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <plus.h>
```

Inheritance diagram for Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Plus\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

## 6.496.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, typename RHS_T>  
class Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
```

Definition at line 140 of file plus.h.

## 6.496.2 Member Typedef Documentation

### 6.496.2.1 LHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Plus_Vector_Vector<  
RTYPE, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 146 of file plus.h.

### 6.496.2.2 LHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef Rcpp::VectorBase<RTYPE, LHS_NA, LHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA,  
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 143 of file plus.h.

### 6.496.2.3 RHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T>::type Rcpp::sugar::Plus_Vector_Vector<  
RTYPE, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 147 of file plus.h.

### 6.496.2.4 RHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 144 of file plus.h.

### 6.496.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::STORAGE
```

Definition at line 142 of file plus.h.

## 6.496.3 Constructor & Destructor Documentation

### 6.496.3.1 Plus\_Vector\_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::Plus_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 149 of file plus.h.

## 6.496.4 Member Function Documentation

### 6.496.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 152 of file plus.h.

References Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.



### 6.496.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 158 of file plus.h.

References Rcpp::sugar::Plus\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs.

## 6.496.5 Member Data Documentation

### 6.496.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::lhs [private]
```

Definition at line 161 of file plus.h.

### 6.496.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::rhs [private]
```

Definition at line 162 of file plus.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/plus.h

## 6.497 Rcpp::sugar::pmax\_op< RTYPE, LHS\_NA, RHS\_NA > Struct Template Reference

### 6.497.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, bool RHS_NA>
struct Rcpp::sugar::pmax_op< RTYPE, LHS_NA, RHS_NA >
```

Definition at line 28 of file pmax.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

## 6.498 Rcpp::sugar::pmax\_op< INTSXP, LHS\_NA, RHS\_NA > Struct Template Reference

```
#include <pmax.h>
```

### Public Member Functions

- int [operator\(\)](#) (int left, int right) const

### 6.498.1 Detailed Description

```
template<bool LHS_NA, bool RHS_NA>  
struct Rcpp::sugar::pmax_op< INTSXP, LHS_NA, RHS_NA >
```

Definition at line 57 of file pmax.h.

### 6.498.2 Member Function Documentation

#### 6.498.2.1 operator>()

```
template<bool LHS_NA, bool RHS_NA>  
int Rcpp::sugar::pmax_op< INTSXP, LHS_NA, RHS_NA >::operator() (  
    int left,  
    int right ) const [inline]
```

Definition at line 58 of file pmax.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmax.h](#)

## 6.499 Rcpp::sugar::pmax\_op< REALSXP, false, false > Struct Reference

```
#include <pmax.h>
```

### Public Member Functions

- double [operator\(\)](#) (double left, double right) const

### 6.499.1 Detailed Description

Definition at line 48 of file pmax.h.

### 6.499.2 Member Function Documentation

#### 6.499.2.1 operator()

```
double Rcpp::sugar::pmax_op< REALSXP, false, false >::operator() (
    double left,
    double right ) const [inline]
```

Definition at line 49 of file pmax.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmax.h](#)

## 6.500 Rcpp::sugar::pmax\_op< REALSXP, false, true > Struct Reference

```
#include <pmax.h>
```

### Public Member Functions

- double [operator\(\)](#) (double left, double right) const

### 6.500.1 Detailed Description

Definition at line 43 of file pmax.h.

### 6.500.2 Member Function Documentation

### 6.500.2.1 operator()

```
double Rcpp::sugar::pmax_op< REALSXP, false, true >::operator() (
    double left,
    double right ) const [inline]
```

Definition at line 44 of file pmax.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmax.h](#)

## 6.501 Rcpp::sugar::pmax\_op< REALSXP, true, false > Struct Reference

```
#include <pmax.h>
```

### Public Member Functions

- double [operator\(\)](#) (double left, double right) const

### 6.501.1 Detailed Description

Definition at line 38 of file pmax.h.

### 6.501.2 Member Function Documentation

#### 6.501.2.1 operator()

```
double Rcpp::sugar::pmax_op< REALSXP, true, false >::operator() (
    double left,
    double right ) const [inline]
```

Definition at line 39 of file pmax.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmax.h](#)

## 6.502 Rcpp::sugar::pmax\_op< REALSXP, true, true > Struct Reference

```
#include <pmax.h>
```

### Public Member Functions

- double [operator\(\)](#) (double left, double right) const

### 6.502.1 Detailed Description

Definition at line 33 of file pmax.h.

### 6.502.2 Member Function Documentation

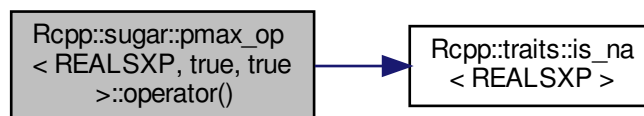
#### 6.502.2.1 operator()

```
double Rcpp::sugar::pmax_op< REALSXP, true, true >::operator() (  
    double left,  
    double right ) const [inline]
```

Definition at line 34 of file pmax.h.

References [Rcpp::traits::is\\_na< REALSXP >\(\)](#).

Here is the call graph for this function:



The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmax.h](#)

## 6.503 Rcpp::sugar::pmax\_op\_Vector\_Primitive< RTYPE, NA > Class Template Reference

```
#include <pmax.h>
```

### Public Types

- typedef [Rcpp::traits::storage\\_type< RTYPE >::type STORAGE](#)

### Public Member Functions

- [pmax\\_op\\_Vector\\_Primitive \(STORAGE right\\_\)](#)
- [STORAGE operator\(\) \(STORAGE left\) const](#)

### Private Attributes

- [STORAGE right](#)

### 6.503.1 Detailed Description

```
template<int RTYPE, bool NA>  
class Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >
```

Definition at line 65 of file pmax.h.

### 6.503.2 Member Typedef Documentation

#### 6.503.2.1 STORAGE

```
template<int RTYPE, bool NA>  
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::pmax\_op\_Vector\_Primitive< RTYPE, NA >::STORAGE
```

Definition at line 67 of file pmax.h.

### 6.503.3 Constructor & Destructor Documentation

### 6.503.3.1 pmax\_op\_Vector\_Primitive()

```
template<int RTYPE, bool NA>
Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >::pmax_op_Vector_Primitive (
    STORAGE right_ ) [inline]
```

Definition at line 69 of file pmax.h.

## 6.503.4 Member Function Documentation

### 6.503.4.1 operator>()()

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >::operator() (
    STORAGE left ) const [inline]
```

Definition at line 71 of file pmax.h.

References Rcpp::sugar::pmax\_op\_Vector\_Primitive< RTYPE, NA >::right.

## 6.503.5 Member Data Documentation

### 6.503.5.1 right

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::pmax_op_Vector_Primitive< RTYPE, NA >::right [private]
```

Definition at line 76 of file pmax.h.

Referenced by Rcpp::sugar::pmax\_op\_Vector\_Primitive< REALSXP, true >::operator()(), and Rcpp::sugar::pmax\_op←\_Vector\_Primitive< RTYPE, NA >::operator()().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmax.h

## 6.504 Rcpp::sugar::pmax\_op\_Vector\_Primitive< REALSXP, true > Class Reference

```
#include <pmax.h>
```

## Public Member Functions

- [pmax\\_op\\_Vector\\_Primitive](#) (double right\_)
- double [operator\(\)](#) (double left) const

## Private Attributes

- double [right](#)

### 6.504.1 Detailed Description

Definition at line 79 of file pmax.h.

### 6.504.2 Constructor & Destructor Documentation

#### 6.504.2.1 pmax\_op\_Vector\_Primitive()

```
Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >::pmax_op_Vector_Primitive (
    double right_ ) [inline]
```

Definition at line 81 of file pmax.h.

### 6.504.3 Member Function Documentation

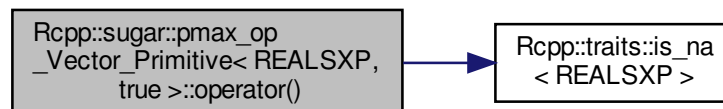
#### 6.504.3.1 operator>()

```
double Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >::operator() (
    double left ) const [inline]
```

Definition at line 83 of file pmax.h.

References [Rcpp::traits::is\\_na< REALSXP >\(\)](#), and [Rcpp::sugar::pmax\\_op\\_Vector\\_Primitive< RTYPE, NA >::right](#).

Here is the call graph for this function:





## 6.504.4 Member Data Documentation

### 6.504.4.1 right

```
double Rcpp::sugar::pmax_op_Vector_Primitive< REALSXP, true >::right [private]
```

Definition at line 88 of file pmax.h.

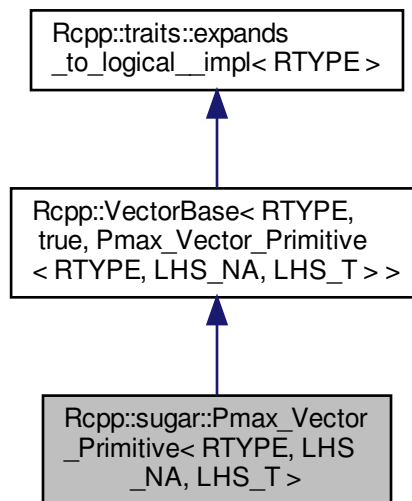
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmax.h](#)

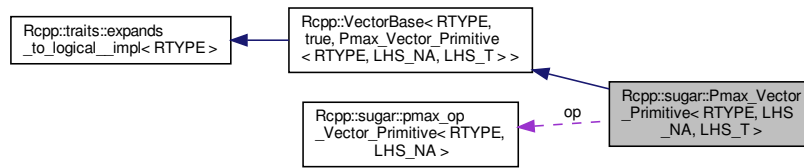
## 6.505 Rcpp::sugar::Pmax\_Vector\_Primitive< RTYPE, LHS\_NA, LHS\_T > Class Template Reference

```
#include <pmax.h>
```

Inheritance diagram for Rcpp::sugar::Pmax\_Vector\_Primitive< RTYPE, LHS\_NA, LHS\_T >:



Collaboration diagram for `Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >`:



## Public Types

- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `pmax_op_Vector_Primitive< RTYPE, LHS_NA >` `OPERATOR`

## Public Member Functions

- `Pmax_Vector_Primitive` (const LHS\_T &lhs\_, `STORAGE` rhs\_)
- `STORAGE operator[]` (R\_xlen\_t i) const
- `R_xlen_t size` () const

## Private Attributes

- const LHS\_T & `lhs`
- `OPERATOR op`

### 6.505.1 Detailed Description

```

template<int RTYPE, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >
  
```

Definition at line 127 of file `pmax.h`.

### 6.505.2 Member Typedef Documentation

### 6.505.2.1 OPERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef pmax_op_Vector_Primitive<RTYPE,LHS_NA> Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA,
LHS_T >::OPERATOR
```

Definition at line 134 of file pmax.h.

### 6.505.2.2 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA,
LHS_T >::STORAGE
```

Definition at line 133 of file pmax.h.

## 6.505.3 Constructor & Destructor Documentation

### 6.505.3.1 Pmax\_Vector\_Primitive()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::Pmax_Vector_Primitive (
    const LHS_T & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 136 of file pmax.h.

## 6.505.4 Member Function Documentation

### 6.505.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
STORAGE Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 138 of file pmax.h.

References [Rcpp::sugar::Pmax\\_Vector\\_Primitive< RTYPE, LHS\\_NA, LHS\\_T >::lhs](#), and [Rcpp::sugar::Pmax\\_Vector\\_Primitive< RTYPE, LHS\\_NA, LHS\\_T >::op](#).

### 6.505.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >  
R_xlen_t Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::size ( ) const [inline]
```

Definition at line 141 of file pmax.h.

References Rcpp::sugar::Pmax\_Vector\_Primitive< RTYPE, LHS\_NA, LHS\_T >::lhs.

## 6.505.5 Member Data Documentation

### 6.505.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T >  
const LHS_T& Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::lhs [private]
```

Definition at line 144 of file pmax.h.

Referenced by Rcpp::sugar::Pmax\_Vector\_Primitive< RTYPE, LHS\_NA, LHS\_T >::operator[](), and Rcpp::sugar::↔ Pmax\_Vector\_Primitive< RTYPE, LHS\_NA, LHS\_T >::size().

### 6.505.5.2 op

```
template<int RTYPE, bool LHS_NA, typename LHS_T >  
OPERATOR Rcpp::sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::op [private]
```

Definition at line 145 of file pmax.h.

Referenced by Rcpp::sugar::Pmax\_Vector\_Primitive< RTYPE, LHS\_NA, LHS\_T >::operator[]().

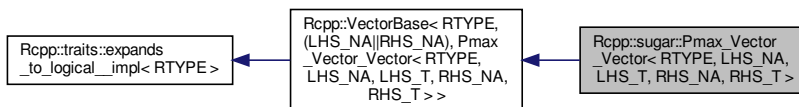
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmax.h](#)

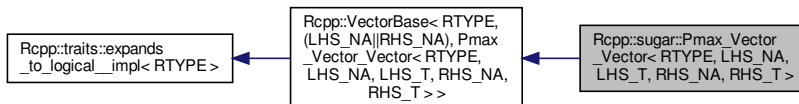
## 6.506 Rcpp::sugar::Pmax\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <pmax.h>
```

Inheritance diagram for Rcpp::sugar::Pmax\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Pmax\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) STORAGE
- typedef [pmax\\_op< RTYPE, LHS\\_NA, RHS\\_NA >](#) OPERATOR

### Public Member Functions

- [Pmax\\_Vector\\_Vector](#) (const LHS\_T &lhs\_, const RHS\_T &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const LHS\_T & lhs
- const RHS\_T & rhs
- [OPERATOR op](#)

### 6.506.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 99 of file pmax.h.

### 6.506.2 Member Typedef Documentation

#### 6.506.2.1 OPERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef pmax_op<RTYPE, LHS_NA, RHS_NA> Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_↵
_NA, RHS_T >::OPERATOR
```

Definition at line 106 of file pmax.h.

#### 6.506.2.2 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 105 of file pmax.h.

### 6.506.3 Constructor & Destructor Documentation

#### 6.506.3.1 Pmax\_Vector\_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Pmax_Vector_Vector (
    const LHS_T & lhs_,
    const RHS_T & rhs_ ) [inline]
```

Definition at line 108 of file pmax.h.

## 6.506.4 Member Function Documentation

### 6.506.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
STORAGE Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (   
    R_xlen_t i ) const [inline]
```

Definition at line 110 of file pmax.h.

References Rcpp::sugar::Pmax\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, Rcpp::sugar::Pmax\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::op, and Rcpp::sugar::Pmax\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.506.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
R_xlen_t Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const  
[inline]
```

Definition at line 113 of file pmax.h.

References Rcpp::sugar::Pmax\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs.

## 6.506.5 Member Data Documentation

### 6.506.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
const LHS_T& Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 116 of file pmax.h.

Referenced by Rcpp::sugar::Pmax\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[](), and Rcpp::sugar::Pmax\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::size().

### 6.506.5.2 op

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
OPERATOR Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::op [private]
```

Definition at line 118 of file pmax.h.

Referenced by Rcpp::sugar::Pmax\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[ ]().

### 6.506.5.3 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_T& Rcpp::sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs [private]
```

Definition at line 117 of file pmax.h.

Referenced by Rcpp::sugar::Pmax\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[ ]().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmax.h](#)

## 6.507 Rcpp::sugar::pmin\_op< RTYPE, LHS\_NA, RHS\_NA > Struct Template Reference

### 6.507.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, bool RHS_NA>
struct Rcpp::sugar::pmin_op< RTYPE, LHS_NA, RHS_NA >
```

Definition at line 28 of file pmin.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmin.h](#)

## 6.508 Rcpp::sugar::pmin\_op< INTSXP, LHS\_NA, RHS\_NA > Struct Template Reference

```
#include <pmin.h>
```



## Public Member Functions

- int [operator\(\)](#) (int left, int right) const

### 6.508.1 Detailed Description

```
template<bool LHS_NA, bool RHS_NA>
struct Rcpp::sugar::pmin_op< INTSXP, LHS_NA, RHS_NA >
```

Definition at line 57 of file pmin.h.

### 6.508.2 Member Function Documentation

#### 6.508.2.1 operator()

```
template<bool LHS_NA, bool RHS_NA>
int Rcpp::sugar::pmin_op< INTSXP, LHS_NA, RHS_NA >::operator() (
    int left,
    int right ) const [inline]
```

Definition at line 58 of file pmin.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmin.h](#)

## 6.509 Rcpp::sugar::pmin\_op< REALSXP, false, false > Struct Reference

```
#include <pmin.h>
```

## Public Member Functions

- double [operator\(\)](#) (double left, double right) const

### 6.509.1 Detailed Description

Definition at line 48 of file pmin.h.

## 6.509.2 Member Function Documentation

### 6.509.2.1 operator()

```
double Rcpp::sugar::pmin_op< REALSXP, false, false >::operator() (  
    double left,  
    double right ) const [inline]
```

Definition at line 49 of file pmin.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmin.h](#)

## 6.510 Rcpp::sugar::pmin\_op< REALSXP, false, true > Struct Reference

```
#include <pmin.h>
```

### Public Member Functions

- double [operator\(\)](#) (double left, double right) const

### 6.510.1 Detailed Description

Definition at line 43 of file pmin.h.

## 6.510.2 Member Function Documentation

### 6.510.2.1 operator()

```
double Rcpp::sugar::pmin_op< REALSXP, false, true >::operator() (  
    double left,  
    double right ) const [inline]
```

Definition at line 44 of file pmin.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmin.h](#)

## 6.511 Rcpp::sugar::pmin\_op< REALSXP, true, false > Struct Reference

```
#include <pmin.h>
```

### Public Member Functions

- double [operator\(\)](#) (double left, double right) const

#### 6.511.1 Detailed Description

Definition at line 38 of file pmin.h.

#### 6.511.2 Member Function Documentation

##### 6.511.2.1 operator()

```
double Rcpp::sugar::pmin_op< REALSXP, true, false >::operator() (  
    double left,  
    double right ) const [inline]
```

Definition at line 39 of file pmin.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

## 6.512 Rcpp::sugar::pmin\_op< REALSXP, true, true > Struct Reference

```
#include <pmin.h>
```

### Public Member Functions

- double [operator\(\)](#) (double left, double right) const

#### 6.512.1 Detailed Description

Definition at line 33 of file pmin.h.

## 6.512.2 Member Function Documentation

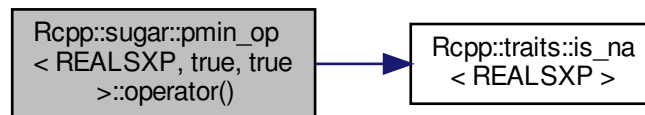
### 6.512.2.1 operator()

```
double Rcpp::sugar::pmin_op< REALSXP, true, true >::operator() (
    double left,
    double right ) const [inline]
```

Definition at line 34 of file pmin.h.

References `Rcpp::traits::is_na< REALSXP >()`.

Here is the call graph for this function:



The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmin.h](#)

## 6.513 Rcpp::sugar::pmin\_op\_Vector\_Primitive< RTYPE, NA > Class Template Reference

```
#include <pmin.h>
```

### Public Types

- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`

### Public Member Functions

- `pmin_op_Vector_Primitive` (`STORAGE` `right_`)
- `STORAGE` `operator()` (`STORAGE` `left`) `const`

## Private Attributes

- [STORAGE right](#)

### 6.513.1 Detailed Description

```
template<int RTYPE, bool NA>  
class Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >
```

Definition at line 65 of file pmin.h.

### 6.513.2 Member Typedef Documentation

#### 6.513.2.1 STORAGE

```
template<int RTYPE, bool NA>  
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA  
>::STORAGE
```

Definition at line 67 of file pmin.h.

### 6.513.3 Constructor & Destructor Documentation

#### 6.513.3.1 pmin\_op\_Vector\_Primitive()

```
template<int RTYPE, bool NA>  
Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >::pmin_op_Vector_Primitive (  
    STORAGE right_ ) [inline]
```

Definition at line 69 of file pmin.h.

### 6.513.4 Member Function Documentation

### 6.513.4.1 operator()

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >::operator() (
    STORAGE left ) const [inline]
```

Definition at line 71 of file pmin.h.

References Rcpp::sugar::pmin\_op\_Vector\_Primitive< RTYPE, NA >::right.

## 6.513.5 Member Data Documentation

### 6.513.5.1 right

```
template<int RTYPE, bool NA>
STORAGE Rcpp::sugar::pmin_op_Vector_Primitive< RTYPE, NA >::right [private]
```

Definition at line 76 of file pmin.h.

Referenced by Rcpp::sugar::pmin\_op\_Vector\_Primitive< REALSXP, true >::operator(), and Rcpp::sugar::pmin\_op←\_Vector\_Primitive< RTYPE, NA >::operator().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmin.h](#)

## 6.514 Rcpp::sugar::pmin\_op\_Vector\_Primitive< REALSXP, true > Class Reference

```
#include <pmin.h>
```

### Public Member Functions

- [pmin\\_op\\_Vector\\_Primitive](#) (double right\_)
- double [operator\(\)](#) (double left) const

### Private Attributes

- double [right](#)

## 6.514.1 Detailed Description

Definition at line 79 of file pmin.h.

## 6.514.2 Constructor & Destructor Documentation

### 6.514.2.1 pmin\_op\_Vector\_Primitive()

```
Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >::pmin_op_Vector_Primitive (
    double right_ ) [inline]
```

Definition at line 81 of file pmin.h.

## 6.514.3 Member Function Documentation

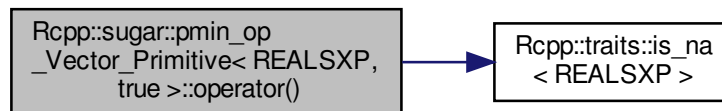
### 6.514.3.1 operator>()

```
double Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >::operator() (
    double left ) const [inline]
```

Definition at line 83 of file pmin.h.

References Rcpp::traits::is\_na< REALSXP >(), and Rcpp::sugar::pmin\_op\_Vector\_Primitive< RTYPE, NA >::right.

Here is the call graph for this function:



## 6.514.4 Member Data Documentation

### 6.514.4.1 right

```
double Rcpp::sugar::pmin_op_Vector_Primitive< REALSXP, true >::right [private]
```

Definition at line 88 of file pmin.h.

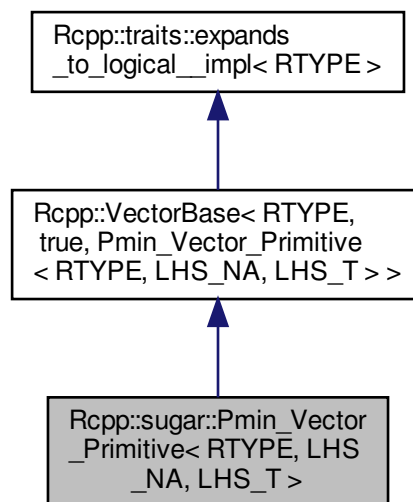
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

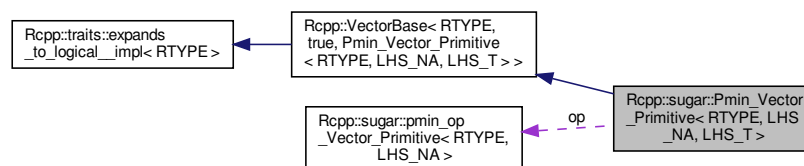
## 6.515 Rcpp::sugar::Pmin\_Vector\_Primitive< RTYPE, LHS\_NA, LHS\_T > Class Template Reference

```
#include <pmin.h>
```

Inheritance diagram for Rcpp::sugar::Pmin\_Vector\_Primitive< RTYPE, LHS\_NA, LHS\_T >:



Collaboration diagram for Rcpp::sugar::Pmin\_Vector\_Primitive< RTYPE, LHS\_NA, LHS\_T >:





## Public Types

- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [pmin\\_op\\_Vector\\_Primitive](#)< RTYPE, LHS\_NA > [OPERATOR](#)

## Public Member Functions

- [Pmin\\_Vector\\_Primitive](#) (const LHS\_T &lhs\_, [STORAGE](#) rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const LHS\_T & [lhs](#)
- [OPERATOR op](#)

### 6.515.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >
```

Definition at line 127 of file pmin.h.

### 6.515.2 Member Typedef Documentation

#### 6.515.2.1 OPERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef pmin\_op\_Vector\_Primitive<RTYPE,LHS_NA> Rcpp::sugar::Pmin\_Vector\_Primitive< RTYPE, LHS_NA,
LHS_T >::OPERATOR
```

Definition at line 134 of file pmin.h.

#### 6.515.2.2 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Pmin\_Vector\_Primitive< RTYPE, LHS_↔
NA, LHS_T >::STORAGE
```

Definition at line 133 of file pmin.h.

### 6.515.3 Constructor & Destructor Documentation

#### 6.515.3.1 Pmin\_Vector\_Primitive()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::Pmin_Vector_Primitive (
    const LHS_T & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 136 of file pmin.h.

### 6.515.4 Member Function Documentation

#### 6.515.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
STORAGE Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 138 of file pmin.h.

References [Rcpp::sugar::Pmin\\_Vector\\_Primitive< RTYPE, LHS\\_NA, LHS\\_T >::lhs](#), and [Rcpp::sugar::Pmin\\_Vector\\_Primitive< RTYPE, LHS\\_NA, LHS\\_T >::op](#).

#### 6.515.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::size ( ) const [inline]
```

Definition at line 139 of file pmin.h.

References [Rcpp::sugar::Pmin\\_Vector\\_Primitive< RTYPE, LHS\\_NA, LHS\\_T >::lhs](#).

### 6.515.5 Member Data Documentation

**6.515.5.1 lhs**

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
const LHS_T& Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::lhs [private]
```

Definition at line 142 of file pmin.h.

Referenced by Rcpp::sugar::Pmin\_Vector\_Primitive< RTYPE, LHS\_NA, LHS\_T >::operator[](), and Rcpp::sugar::Pmin\_Vector\_Primitive< RTYPE, LHS\_NA, LHS\_T >::size().

**6.515.5.2 op**

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
OPERATOR Rcpp::sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >::op [private]
```

Definition at line 143 of file pmin.h.

Referenced by Rcpp::sugar::Pmin\_Vector\_Primitive< RTYPE, LHS\_NA, LHS\_T >::operator[]().

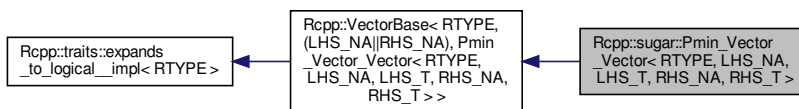
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pmin.h

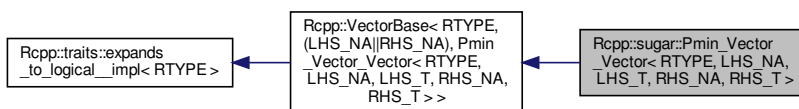
**6.516 Rcpp::sugar::Pmin\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference**

```
#include <pmin.h>
```

Inheritance diagram for Rcpp::sugar::Pmin\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Pmin\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



## Public Types

- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [pmin\\_op](#)< RTYPE, LHS\_NA, RHS\_NA > [OPERATOR](#)

## Public Member Functions

- [Pmin\\_Vector\\_Vector](#) (const LHS\_T &lhs\_, const RHS\_T &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const LHS\_T & [lhs](#)
- const RHS\_T & [rhs](#)
- [OPERATOR op](#)

### 6.516.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 99 of file pmin.h.

### 6.516.2 Member Typedef Documentation

#### 6.516.2.1 OPERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef pmin\_op<RTYPE,LHS_NA,RHS_NA> Rcpp::sugar::Pmin\_Vector\_Vector< RTYPE, LHS_NA, LHS_T, RHS←
_NA, RHS_T >::OPERATOR
```

Definition at line 106 of file pmin.h.

#### 6.516.2.2 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Pmin\_Vector\_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::STORAGE
```

Definition at line 105 of file pmin.h.

## 6.516.3 Constructor & Destructor Documentation

### 6.516.3.1 Pmin\_Vector\_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Pmin_Vector_Vector (  
    const LHS_T & lhs_,  
    const RHS_T & rhs_ ) [inline]
```

Definition at line 108 of file pmin.h.

## 6.516.4 Member Function Documentation

### 6.516.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
STORAGE Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (  
    R_xlen_t i ) const [inline]
```

Definition at line 110 of file pmin.h.

References Rcpp::sugar::Pmin\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, Rcpp::sugar::Pmin\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::op, and Rcpp::sugar::Pmin\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.516.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
R_xlen_t Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const  
[inline]
```

Definition at line 113 of file pmin.h.

References Rcpp::sugar::Pmin\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs.

## 6.516.5 Member Data Documentation

### 6.516.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_T& Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 116 of file pmin.h.

Referenced by `Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[]()`, and `Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size()`.

### 6.516.5.2 op

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
OPERATOR Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::op [private]
```

Definition at line 118 of file pmin.h.

Referenced by `Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[]()`.

### 6.516.5.3 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_T& Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs [private]
```

Definition at line 117 of file pmin.h.

Referenced by `Rcpp::sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[]()`.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pmin.h](#)

## 6.517 Rcpp::traits::pointer\_wrap\_tag Struct Reference

```
#include <module_wrap_traits.h>
```

### 6.517.1 Detailed Description

Definition at line 31 of file `module_wrap_traits.h`.

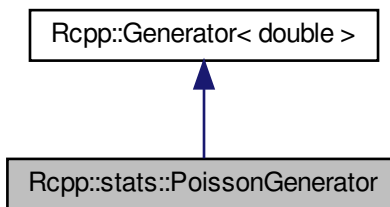
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/module\\_wrap\\_traits.h](#)

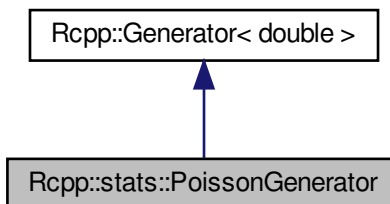
## 6.518 Rcpp::stats::PoissonGenerator Class Reference

```
#include <rpois.h>
```

Inheritance diagram for Rcpp::stats::PoissonGenerator:



Collaboration diagram for Rcpp::stats::PoissonGenerator:



### Public Member Functions

- [PoissonGenerator](#) (double mu\_)
- double [operator\(\)](#) () const

### Private Attributes

- double [mu](#)

## Additional Inherited Members

### 6.518.1 Detailed Description

Definition at line 28 of file rpois.h.

### 6.518.2 Constructor & Destructor Documentation

#### 6.518.2.1 PoissonGenerator()

```
Rcpp::stats::PoissonGenerator::PoissonGenerator (  
    double mu_ ) [inline]
```

Definition at line 30 of file rpois.h.

### 6.518.3 Member Function Documentation

#### 6.518.3.1 operator>()()

```
double Rcpp::stats::PoissonGenerator::operator() ( ) const [inline]
```

Definition at line 31 of file rpois.h.

References mu.

### 6.518.4 Member Data Documentation

#### 6.518.4.1 mu

```
double Rcpp::stats::PoissonGenerator::mu [private]
```

Definition at line 33 of file rpois.h.

Referenced by operator>()().

The documentation for this class was generated from the following file:

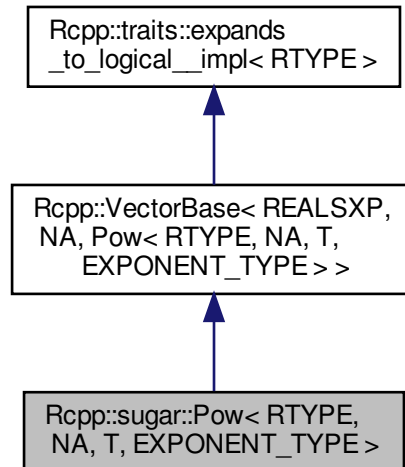
- [inst/include/Rcpp/stats/random/rpois.h](#)



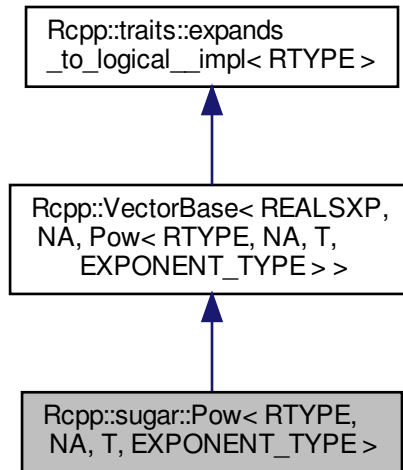
## 6.519 Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT\_TYPE > Class Template Reference

```
#include <pow.h>
```

Inheritance diagram for Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT\_TYPE >:



Collaboration diagram for `Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >`:



## Public Types

- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`

## Public Member Functions

- `Pow` (const T &object\_, EXPONENT\_TYPE exponent)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const T & `object`
- EXPONENT\_TYPE `op`

### 6.519.1 Detailed Description

```

template<int RTYPE, bool NA, typename T, typename EXPONENT_TYPE>
class Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >
  
```

Definition at line 29 of file `pow.h`.

## 6.519.2 Member Typedef Documentation

### 6.519.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE
>::STORAGE
```

Definition at line 31 of file pow.h.

## 6.519.3 Constructor & Destructor Documentation

### 6.519.3.1 Pow()

```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >
Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::Pow (
    const T & object_,
    EXPONENT_TYPE exponent ) [inline]
```

Definition at line 33 of file pow.h.

## 6.519.4 Member Function Documentation

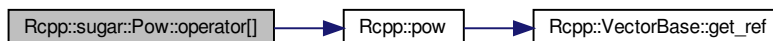
### 6.519.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >
double Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 35 of file pow.h.

References Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT\_TYPE >::op, and Rcpp::pow().

Here is the call graph for this function:



### 6.519.4.2 size()

```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >  
R_xlen_t Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::size ( ) const [inline]
```

Definition at line 38 of file pow.h.

## 6.519.5 Member Data Documentation

### 6.519.5.1 object

```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >  
const T& Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::object [private]
```

Definition at line 41 of file pow.h.

### 6.519.5.2 op

```
template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >  
EXPONENT_TYPE Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::op [private]
```

Definition at line 42 of file pow.h.

Referenced by `Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::operator[]()`, `Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >::operator[]()`, and `Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >::operator[]()`.

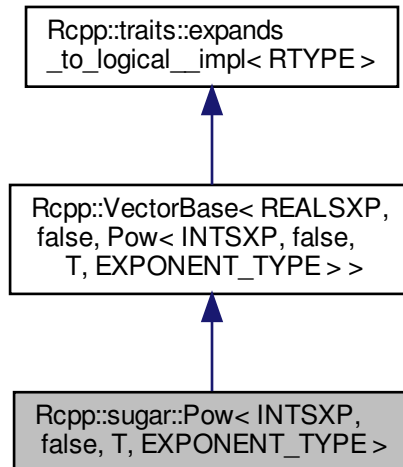
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pow.h](#)

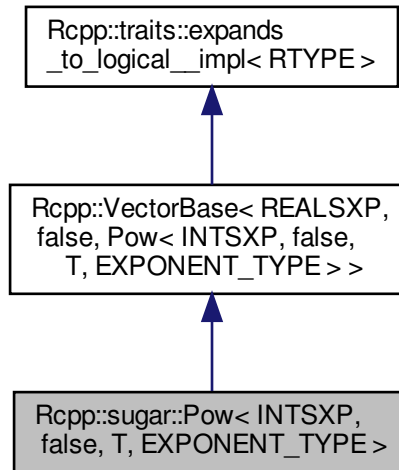
## 6.520 Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT\_TYPE > Class Template Reference

```
#include <pow.h>
```

Inheritance diagram for Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT\_TYPE >:



Collaboration diagram for `Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >`:



## Public Member Functions

- `Pow` (const T &object\_, EXPONENT\_TYPE exponent)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const T & `object`
- EXPONENT\_TYPE `op`

## Additional Inherited Members

### 6.520.1 Detailed Description

```

template<typename T, typename EXPONENT_TYPE>
class Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >
  
```

Definition at line 61 of file pow.h.

### 6.520.2 Constructor & Destructor Documentation

### 6.520.2.1 Pow()

```
template<typename T , typename EXPONENT_TYPE >
Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >::Pow (
    const T & object_,
    EXPONENT_TYPE exponent ) [inline]
```

Definition at line 63 of file pow.h.

## 6.520.3 Member Function Documentation

### 6.520.3.1 operator[]()

```
template<typename T , typename EXPONENT_TYPE >
double Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 65 of file pow.h.

References Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT\_TYPE >::op, and Rcpp::pow().

Here is the call graph for this function:



### 6.520.3.2 size()

```
template<typename T , typename EXPONENT_TYPE >
R_xlen_t Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >::size ( ) const [inline]
```

Definition at line 68 of file pow.h.

## 6.520.4 Member Data Documentation

### 6.520.4.1 object

```
template<typename T , typename EXPONENT_TYPE >
const T& Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >::object [private]
```

Definition at line 71 of file pow.h.

### 6.520.4.2 op

```
template<typename T , typename EXPONENT_TYPE >
EXPONENT_TYPE Rcpp::sugar::Pow< INTSXP, false, T, EXPONENT_TYPE >::op [private]
```

Definition at line 72 of file pow.h.

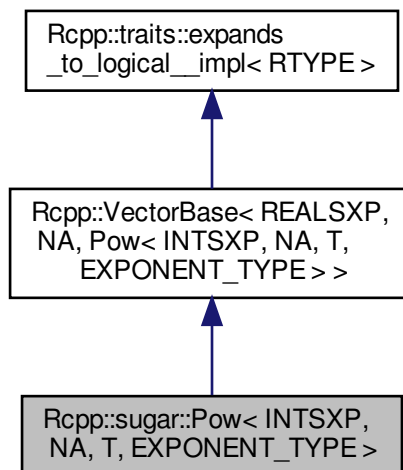
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/pow.h

## 6.521 Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT\_TYPE > Class Template Reference

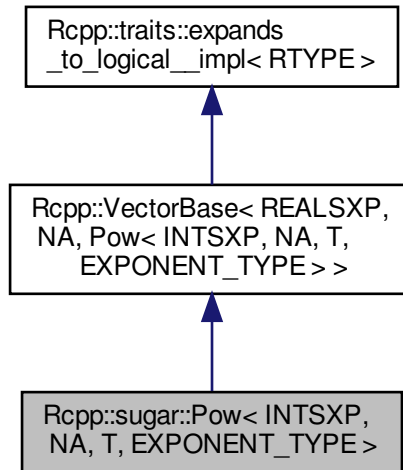
```
#include <pow.h>
```

Inheritance diagram for Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT\_TYPE >:





Collaboration diagram for Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT\_TYPE >:



## Public Member Functions

- `Pow` (const T &object\_, EXPONENT\_TYPE exponent)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const T & `object`
- EXPONENT\_TYPE `op`

## Additional Inherited Members

### 6.521.1 Detailed Description

```

template<bool NA, typename T, typename EXPONENT_TYPE>
class Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >
  
```

Definition at line 46 of file pow.h.

### 6.521.2 Constructor & Destructor Documentation

### 6.521.2.1 Pow()

```
template<bool NA, typename T , typename EXPONENT_TYPE >
Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >::Pow (
    const T & object_,
    EXPONENT_TYPE exponent ) [inline]
```

Definition at line 48 of file pow.h.

## 6.521.3 Member Function Documentation

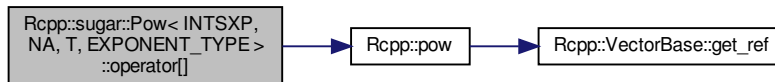
### 6.521.3.1 operator[]()

```
template<bool NA, typename T , typename EXPONENT_TYPE >
double Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 50 of file pow.h.

References `Rcpp::sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >::op`, and `Rcpp::pow()`.

Here is the call graph for this function:



### 6.521.3.2 size()

```
template<bool NA, typename T , typename EXPONENT_TYPE >
R_xlen_t Rcpp::sugar::Pow< INTSXP, NA, T, EXPONENT_TYPE >::size ( ) const [inline]
```

Definition at line 54 of file pow.h.

## 6.521.4 Member Data Documentation

### 6.521.4.1 object

```
template<bool NA, typename T , typename EXPONENT_TYPE >  
const T& Rcpp::sugar::Pow< INTXP, NA, T, EXPONENT_TYPE >::object [private]
```

Definition at line 57 of file pow.h.

### 6.521.4.2 op

```
template<bool NA, typename T , typename EXPONENT_TYPE >  
EXPONENT_TYPE Rcpp::sugar::Pow< INTXP, NA, T, EXPONENT_TYPE >::op [private]
```

Definition at line 58 of file pow.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/pow.h](#)

## 6.522 Rcpp::PreserveStorage< CLASS > Class Template Reference

```
#include <PreserveStorage.h>
```

### Public Member Functions

- [PreserveStorage](#) ()
- [~PreserveStorage](#) ()
- void [set\\_\\_](#) (SEXP x)
- SEXP [get\\_\\_](#) () const
- SEXP [invalidate\\_\\_](#) ()
- template<typename T >  
T & [copy\\_\\_](#) (const T &other)
- bool [inherits](#) (const char \*clazz) const
- [operator SEXP](#) () const

### Private Attributes

- SEXP [data](#)
- SEXP [token](#)

### 6.522.1 Detailed Description

```
template<typename CLASS>
class Rcpp::PreserveStorage< CLASS >
```

Definition at line 28 of file PreserveStorage.h.

### 6.522.2 Constructor & Destructor Documentation

#### 6.522.2.1 PreserveStorage()

```
template<typename CLASS >
Rcpp::PreserveStorage< CLASS >::PreserveStorage ( ) [inline]
```

Definition at line 31 of file PreserveStorage.h.

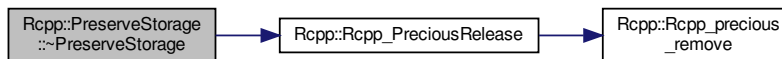
#### 6.522.2.2 ~PreserveStorage()

```
template<typename CLASS >
Rcpp::PreserveStorage< CLASS >::~~PreserveStorage ( ) [inline]
```

Definition at line 33 of file PreserveStorage.h.

References `Rcpp::PreserveStorage< CLASS >::data`, `Rcpp::Rcpp_PreciousRelease()`, and `Rcpp::PreserveStorage< CLASS >::token`.

Here is the call graph for this function:



### 6.522.3 Member Function Documentation

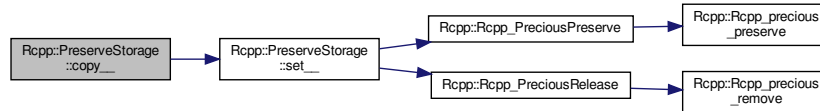
### 6.522.3.1 copy\_\_()

```
template<typename CLASS >
template<typename T >
T& Rcpp::PreserveStorage< CLASS >::copy__ (
    const T & other ) [inline]
```

Definition at line 64 of file PreserveStorage.h.

References Rcpp::PreserveStorage< CLASS >::set\_\_().

Here is the call graph for this function:



### 6.522.3.2 get\_\_()

```
template<typename CLASS >
SEXPR Rcpp::PreserveStorage< CLASS >::get__ ( ) const [inline]
```

Definition at line 51 of file PreserveStorage.h.

References Rcpp::PreserveStorage< CLASS >::data.

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::erase\_range\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::erase\_single\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::insert\_\_impl(), Rcpp::Matrix< RTYPE, StoragePolicy >::operator=(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_back\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_back\_name\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_front\_\_impl(), and Rcpp::Vector< RTYPE, StoragePolicy >::push\_front\_name\_\_impl().

### 6.522.3.3 inherits()

```
template<typename CLASS >
bool Rcpp::PreserveStorage< CLASS >::inherits (
    const char * clazz ) const [inline]
```

Definition at line 71 of file PreserveStorage.h.

References Rcpp::PreserveStorage< CLASS >::data.

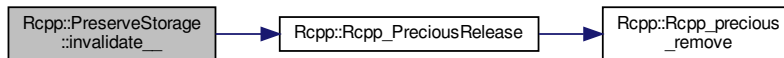
### 6.522.3.4 invalidate\_\_()

```
template<typename CLASS >
SEXP Rcpp::PreserveStorage< CLASS >::invalidate__ ( ) [inline]
```

Definition at line 55 of file PreserveStorage.h.

References Rcpp::PreserveStorage< CLASS >::data, Rcpp::Rcpp\_PreciousRelease(), and Rcpp::PreserveStorage< CLASS >::token.

Here is the call graph for this function:



### 6.522.3.5 operator SEXP()

```
template<typename CLASS >
Rcpp::PreserveStorage< CLASS >::operator SEXP ( ) const [inline]
```

Definition at line 75 of file PreserveStorage.h.

References Rcpp::PreserveStorage< CLASS >::data.

### 6.522.3.6 set\_\_()

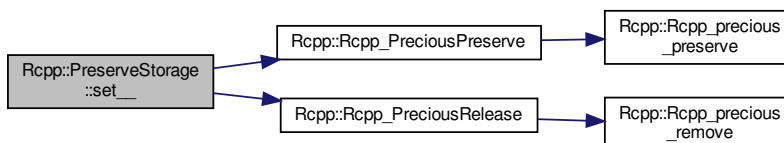
```
template<typename CLASS >
void Rcpp::PreserveStorage< CLASS >::set__ (
    SEXP x ) [inline]
```

Definition at line 39 of file PreserveStorage.h.

References Rcpp::PreserveStorage< CLASS >::data, Rcpp::Rcpp\_PreciousPreserve(), Rcpp::Rcpp\_PreciousRelease(), and Rcpp::PreserveStorage< CLASS >::token.

Referenced by Rcpp::PreserveStorage< CLASS >::copy\_\_().

Here is the call graph for this function:



## 6.522.4 Member Data Documentation

### 6.522.4.1 data

```
template<typename CLASS >
SEXPRcpp::PreserveStorage< CLASS >::data [private]
```

Definition at line 78 of file PreserveStorage.h.

Referenced by Rcpp::PreserveStorage< CLASS >::get\_\_(), Rcpp::PreserveStorage< CLASS >::inherits(), Rcpp::PreserveStorage< CLASS >::invalidate\_\_(), Rcpp::PreserveStorage< CLASS >::operator SEXP(), Rcpp::PreserveStorage< CLASS >::set\_\_(), and Rcpp::PreserveStorage< CLASS >::~~PreserveStorage().

### 6.522.4.2 token

```
template<typename CLASS >
SEXPRcpp::PreserveStorage< CLASS >::token [private]
```

Definition at line 79 of file PreserveStorage.h.

Referenced by Rcpp::PreserveStorage< CLASS >::invalidate\_\_(), Rcpp::PreserveStorage< CLASS >::set\_\_(), and Rcpp::PreserveStorage< CLASS >::~~PreserveStorage().

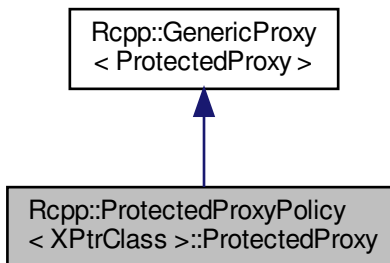
The documentation for this class was generated from the following file:

- inst/include/Rcpp/storage/PreserveStorage.h

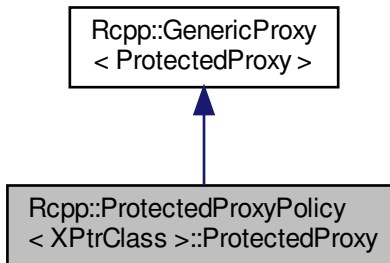
## 6.523 Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy Class Reference

```
#include <ProtectedProxy.h>
```

Inheritance diagram for Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy:



Collaboration diagram for Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy:



## Public Member Functions

- [ProtectedProxy](#) (XPtrClass &xp\_)
- template<typename U >  
[ProtectedProxy](#) & [operator=](#) (const U &u)
- template<typename U >  
[operator U](#) () const
- [operator SEXP](#) () const

## Private Member Functions

- SEXP [get](#) () const
- void [set](#) (SEXP x)

## Private Attributes

- XPtrClass & [xp](#)

### 6.523.1 Detailed Description

```

template<class XPtrClass>
class Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy
  
```

Definition at line 29 of file ProtectedProxy.h.

### 6.523.2 Constructor & Destructor Documentation



### 6.523.2.1 ProtectedProxy()

```
template<class XPtrClass >  
Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::ProtectedProxy (  
    XPtrClass & xp_ ) [inline]
```

Definition at line 31 of file ProtectedProxy.h.

## 6.523.3 Member Function Documentation

### 6.523.3.1 get()

```
template<class XPtrClass >  
SEXPR Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::get ( ) const [inline], [private]
```

Definition at line 51 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::xp.

Referenced by Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator SEXP(), and Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator U().

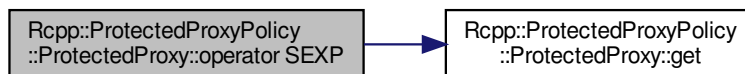
### 6.523.3.2 operator SEXP()

```
template<class XPtrClass >  
Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator SEXP ( ) const [inline]
```

Definition at line 44 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::get().

Here is the call graph for this function:



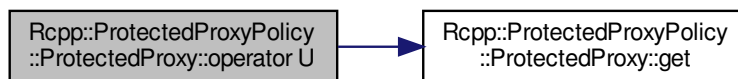
### 6.523.3.3 operator U()

```
template<class XPtrClass >
template<typename U >
Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator U ( ) const [inline]
```

Definition at line 40 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::get().

Here is the call graph for this function:



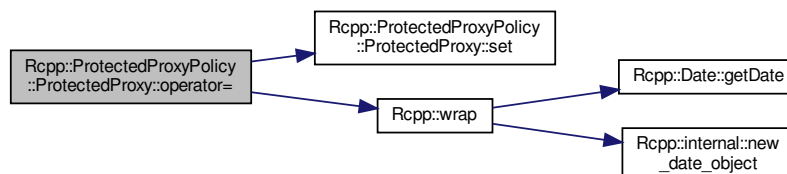
### 6.523.3.4 operator=()

```
template<class XPtrClass >
template<typename U >
ProtectedProxy& Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator= (
    const U & u ) [inline]
```

Definition at line 34 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::set(), and Rcpp::wrap().

Here is the call graph for this function:



### 6.523.3.5 set()

```
template<class XPtrClass >
void Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::set (
    SEXP x ) [inline], [private]
```

Definition at line 55 of file ProtectedProxy.h.

References Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::xp.

Referenced by Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::operator=().

## 6.523.4 Member Data Documentation

### 6.523.4.1 xp

```
template<class XPtrClass >
XPtrClass& Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::xp [private]
```

Definition at line 49 of file ProtectedProxy.h.

Referenced by Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::get(), and Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy::set().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/ProtectedProxy.h

## 6.524 Rcpp::ProtectedProxyPolicy< XPtrClass > Class Template Reference

```
#include <ProtectedProxy.h>
```

### Classes

- class [const\\_ProtectedProxy](#)
- class [ProtectedProxy](#)

### Public Member Functions

- [ProtectedProxy prot \(\)](#)
- [const\\_ProtectedProxy prot \(\) const](#)

### 6.524.1 Detailed Description

```
template<class XPtrClass>
class Rcpp::ProtectedProxyPolicy< XPtrClass >
```

Definition at line 26 of file ProtectedProxy.h.

### 6.524.2 Member Function Documentation

#### 6.524.2.1 prot() [1/2]

```
template<class XPtrClass >
ProtectedProxy Rcpp::ProtectedProxyPolicy< XPtrClass >::prot ( ) [inline]
```

Definition at line 83 of file ProtectedProxy.h.

#### 6.524.2.2 prot() [2/2]

```
template<class XPtrClass >
const_ProtectedProxy Rcpp::ProtectedProxyPolicy< XPtrClass >::prot ( ) const [inline]
```

Definition at line 87 of file ProtectedProxy.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/ProtectedProxy.h

## 6.525 Rcpp::traits::proxy\_based\_const\_iterator< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef ::Rcpp::internal::Proxy\_Iterator< typename [r\\_vector\\_const\\_proxy](#)< RTYPE, StoragePolicy >::type >  
[type](#)

### 6.525.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>
struct Rcpp::traits::proxy_based_const_iterator< RTYPE, StoragePolicy >
```

Definition at line 267 of file proxy.h.

### 6.525.2 Member Typedef Documentation

#### 6.525.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::internal::Proxy_Iterator< typename r_vector_const_proxy<RTYPE, StoragePolicy>&←
::type > Rcpp::traits::proxy_based_const_iterator< RTYPE, StoragePolicy >::type
```

Definition at line 268 of file proxy.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

## 6.526 Rcpp::traits::proxy\_based\_iterator< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef ::Rcpp::internal::Proxy\_Iterator< typename r\_vector\_proxy< RTYPE, StoragePolicy >::type > type

### 6.526.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>
struct Rcpp::traits::proxy_based_iterator< RTYPE, StoragePolicy >
```

Definition at line 259 of file proxy.h.

## 6.526.2 Member Typedef Documentation

### 6.526.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::internal::Proxy_Iterator< typename r\_vector\_proxy<RTYPE, StoragePolicy>::type >
Rcpp::traits::proxy\_based\_iterator< RTYPE, StoragePolicy >::type
```

Definition at line 260 of file proxy.h.

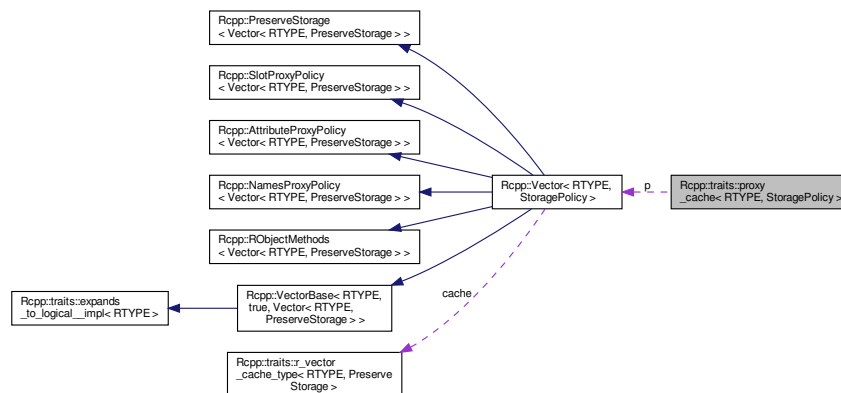
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

## 6.527 Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy > Class Template Reference

```
#include <traits.h>
```

Collaboration diagram for `Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >`:



### Public Types

- typedef `::Rcpp::Vector< RTYPE, StoragePolicy >` `VECTOR`
- typedef `r_vector_iterator< RTYPE, StoragePolicy >::type` `iterator`
- typedef `r_vector_const_iterator< RTYPE, StoragePolicy >::type` `const_iterator`
- typedef `r_vector_proxy< RTYPE, StoragePolicy >::type` `proxy`
- typedef `r_vector_const_proxy< RTYPE, StoragePolicy >::type` `const_proxy`

## Public Member Functions

- [proxy\\_cache](#) ()
- [~proxy\\_cache](#) ()
- void [update](#) (const [VECTOR](#) &v)
- [iterator get](#) () const
- [const\\_iterator get\\_const](#) () const
- [proxy ref](#) ()
- [proxy ref](#) (R\_xlen\_t i)
- [const\\_proxy ref](#) () const
- [const\\_proxy ref](#) (R\_xlen\_t i) const

## Private Attributes

- [VECTOR](#) \* p

### 6.527.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
class Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >
```

Definition at line 55 of file traits.h.

### 6.527.2 Member Typedef Documentation

#### 6.527.2.1 const\_iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r\_vector\_const\_iterator<RTYPE, StoragePolicy>::type Rcpp::traits::proxy\_cache< RTYPE,
StoragePolicy >::const_iterator
```

Definition at line 59 of file traits.h.

#### 6.527.2.2 const\_proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r\_vector\_const\_proxy<RTYPE, StoragePolicy>::type Rcpp::traits::proxy\_cache< RTYPE, Storage←
Policy >::const_proxy
```

Definition at line 61 of file traits.h.

### 6.527.2.3 iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r\_vector\_iterator<RTYPE, StoragePolicy>::type Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::iterator
```

Definition at line 58 of file traits.h.

### 6.527.2.4 proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r\_vector\_proxy<RTYPE, StoragePolicy>::type Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::proxy
```

Definition at line 60 of file traits.h.

### 6.527.2.5 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::VECTOR
```

Definition at line 57 of file traits.h.

## 6.527.3 Constructor & Destructor Documentation

### 6.527.3.1 proxy\_cache()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::proxy_cache ( ) [inline]
```

Definition at line 63 of file traits.h.

### 6.527.3.2 ~proxy\_cache()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::~proxy_cache ( ) [inline]
```

Definition at line 64 of file traits.h.



## 6.527.4 Member Function Documentation

### 6.527.4.1 get()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>  
iterator Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::get ( ) const [inline]
```

Definition at line 68 of file traits.h.

References Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::p.

### 6.527.4.2 get\_const()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>  
const_iterator Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::get_const ( ) const [inline]
```

Definition at line 70 of file traits.h.

References Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::p.

### 6.527.4.3 ref() [1/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>  
proxy Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::ref ( ) [inline]
```

Definition at line 72 of file traits.h.

References Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::p.

### 6.527.4.4 ref() [2/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>  
const_proxy Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::ref ( ) const [inline]
```

Definition at line 75 of file traits.h.

References Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::p.

**6.527.4.5 ref()** [3/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
proxy Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::ref (
    R_xlen_t i ) [inline]
```

Definition at line 73 of file traits.h.

References Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::p.

**6.527.4.6 ref()** [4/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_proxy Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::ref (
    R_xlen_t i ) const [inline]
```

Definition at line 76 of file traits.h.

References Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::p.

**6.527.4.7 update()**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::update (
    const VECTOR & v ) [inline]
```

Definition at line 65 of file traits.h.

References Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::p.

**6.527.5 Member Data Documentation****6.527.5.1 p**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
VECTOR* Rcpp::traits::proxy_cache< RTYPE, StoragePolicy >::p [private]
```

Definition at line 79 of file traits.h.

Referenced by Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::get(), Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::get\_const(), Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::ref(), and Rcpp::traits::proxy\_cache< RTYPE, StoragePolicy >::update().

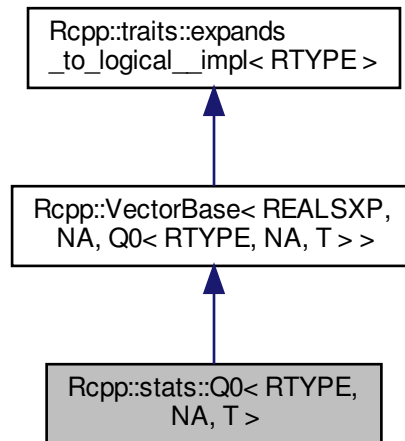
The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/traits.h

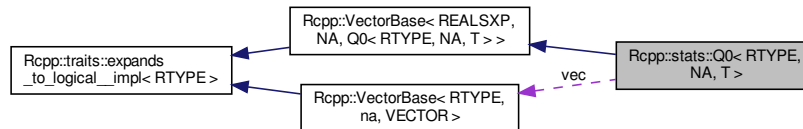
## 6.528 Rcpp::stats::Q0< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::Q0< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::Q0< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `double(* FunPtr) (double, int, int)`

### Public Member Functions

- `Q0 (FunPtr ptr_, const VEC_TYPE &vec_, bool lower_tail=true, bool log_=false)`
- `double operator[] (int i) const`
- `int size () const`

## Private Attributes

- [FunPtr ptr](#)
- const [VEC\\_TYPE](#) & [vec](#)
- int [lower](#)
- int [log](#)

### 6.528.1 Detailed Description

```
template<int RTYPE, bool NA, typename T >  
class Rcpp::stats::Q0< RTYPE, NA, T >
```

Definition at line 221 of file dpq.h.

### 6.528.2 Member Typedef Documentation

#### 6.528.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >  
typedef double(* Rcpp::stats::Q0< RTYPE, NA, T >::FunPtr) (double, int, int)
```

Definition at line 224 of file dpq.h.

#### 6.528.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::Q0< RTYPE, NA, T >::VEC\_TYPE
```

Definition at line 223 of file dpq.h.

### 6.528.3 Constructor & Destructor Documentation

### 6.528.3.1 Q0()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::Q0< RTYPE, NA, T >::Q0 (
    FunPtr ptr_,
    const VEC_TYPE & vec_,
    bool lower_tail = true,
    bool log_ = false ) [inline]
```

Definition at line 226 of file dpq.h.

## 6.528.4 Member Function Documentation

### 6.528.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q0< RTYPE, NA, T >::operator[] (
    int i ) const [inline]
```

Definition at line 230 of file dpq.h.

References Rcpp::stats::Q0< RTYPE, NA, T >::log, Rcpp::stats::Q0< RTYPE, NA, T >::lower, Rcpp::stats::Q0< RTYPE, NA, T >::ptr, and Rcpp::stats::Q0< RTYPE, NA, T >::vec.

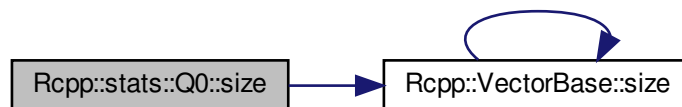
### 6.528.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q0< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 234 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::Q0< RTYPE, NA, T >::vec.

Here is the call graph for this function:



## 6.528.5 Member Data Documentation

### 6.528.5.1 log

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::Q0< RTYPE, NA, T >::log [private]
```

Definition at line 239 of file dpq.h.

Referenced by Rcpp::stats::Q0< RTYPE, NA, T >::operator[]().

### 6.528.5.2 lower

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::Q0< RTYPE, NA, T >::lower [private]
```

Definition at line 239 of file dpq.h.

Referenced by Rcpp::stats::Q0< RTYPE, NA, T >::operator[]().

### 6.528.5.3 ptr

```
template<int RTYPE, bool NA, typename T >  
FunPtr Rcpp::stats::Q0< RTYPE, NA, T >::ptr [private]
```

Definition at line 237 of file dpq.h.

Referenced by Rcpp::stats::Q0< RTYPE, NA, T >::operator[]().

### 6.528.5.4 vec

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::stats::Q0< RTYPE, NA, T >::vec [private]
```

Definition at line 238 of file dpq.h.

Referenced by Rcpp::stats::Q0< RTYPE, NA, T >::operator[](), and Rcpp::stats::Q0< RTYPE, NA, T >::size().

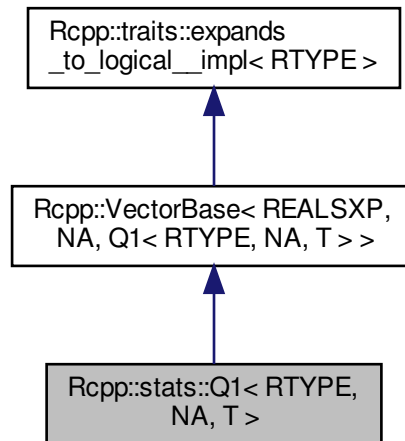
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

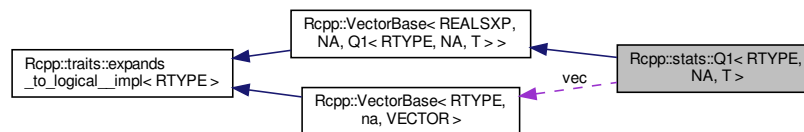
## 6.529 Rcpp::stats::Q1< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::Q1< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::Q1< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `double(* FunPtr) (double, double, int, int)`

### Public Member Functions

- `Q1 (FunPtr ptr_, const VEC_TYPE &vec_, double p0_, bool lower_tail=true, bool log_=false)`
- `double operator[] (int i) const`
- `int size () const`

## Private Attributes

- [FunPtr ptr](#)
- const [VEC\\_TYPE](#) & [vec](#)
- double [p0](#)
- int [lower](#)
- int [log](#)

### 6.529.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::stats::Q1< RTYPE, NA, T >
```

Definition at line 244 of file dpq.h.

### 6.529.2 Member Typedef Documentation

#### 6.529.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >  
typedef double(* Rcpp::stats::Q1< RTYPE, NA, T >::FunPtr) (double, double, int, int)
```

Definition at line 247 of file dpq.h.

#### 6.529.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::Q1< RTYPE, NA, T >::VEC\\_TYPE
```

Definition at line 246 of file dpq.h.

### 6.529.3 Constructor & Destructor Documentation



### 6.529.3.1 Q1()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::Q1< RTYPE, NA, T >::Q1 (
    FunPtr ptr_,
    const VEC_TYPE & vec_,
    double p0_,
    bool lower_tail = true,
    bool log_ = false ) [inline]
```

Definition at line 249 of file dpq.h.

## 6.529.4 Member Function Documentation

### 6.529.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q1< RTYPE, NA, T >::operator[] (
    int i ) const [inline]
```

Definition at line 253 of file dpq.h.

References `Rcpp::stats::Q1< RTYPE, NA, T >::log`, `Rcpp::stats::Q1< RTYPE, NA, T >::lower`, `Rcpp::stats::Q1< RTYPE, NA, T >::p0`, `Rcpp::stats::Q1< RTYPE, NA, T >::ptr`, and `Rcpp::stats::Q1< RTYPE, NA, T >::vec`.

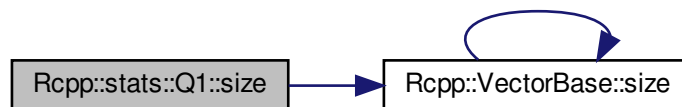
### 6.529.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q1< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 257 of file dpq.h.

References `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`, and `Rcpp::stats::Q1< RTYPE, NA, T >::vec`.

Here is the call graph for this function:



## 6.529.5 Member Data Documentation

### 6.529.5.1 log

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::Q1< RTYPE, NA, T >::log [private]
```

Definition at line 263 of file dpq.h.

Referenced by Rcpp::stats::Q1< RTYPE, NA, T >::operator[]().

### 6.529.5.2 lower

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::Q1< RTYPE, NA, T >::lower [private]
```

Definition at line 263 of file dpq.h.

Referenced by Rcpp::stats::Q1< RTYPE, NA, T >::operator[]().

### 6.529.5.3 p0

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::Q1< RTYPE, NA, T >::p0 [private]
```

Definition at line 262 of file dpq.h.

Referenced by Rcpp::stats::Q1< RTYPE, NA, T >::operator[]().

### 6.529.5.4 ptr

```
template<int RTYPE, bool NA, typename T >  
FunPtr Rcpp::stats::Q1< RTYPE, NA, T >::ptr [private]
```

Definition at line 260 of file dpq.h.

Referenced by Rcpp::stats::Q1< RTYPE, NA, T >::operator[]().

## 6.529.5.5 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::Q1< RTYPE, NA, T >::vec [private]
```

Definition at line 261 of file dpq.h.

Referenced by Rcpp::stats::Q1< RTYPE, NA, T >::operator[](), and Rcpp::stats::Q1< RTYPE, NA, T >::size().

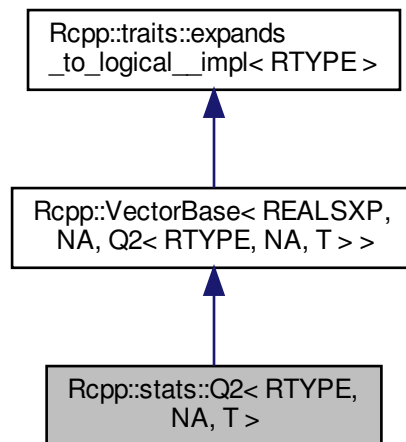
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/dpq/dpq.h

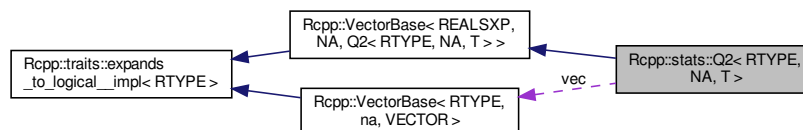
## 6.530 Rcpp::stats::Q2&lt; RTYPE, NA, T &gt; Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::Q2< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::Q2< RTYPE, NA, T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, NA, T > [VEC\\_TYPE](#)
- typedef double(\* [FunPtr](#)) (double, double, double, int, int)

## Public Member Functions

- [Q2](#) ([FunPtr](#) ptr\_, const [VEC\\_TYPE](#) &vec\_, double p0\_, double p1\_, bool lower\_tail=true, bool log\_=false)
- double [operator\[\]](#) (int i) const
- int [size](#) () const

## Private Attributes

- [FunPtr](#) ptr
- const [VEC\\_TYPE](#) & [vec](#)
- double [p0](#)
- double [p1](#)
- int [lower](#)
- int [log](#)

### 6.530.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::stats::Q2< RTYPE, NA, T >
```

Definition at line 268 of file dpq.h.

### 6.530.2 Member Typedef Documentation

#### 6.530.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >  
typedef double(* Rcpp::stats::Q2< RTYPE, NA, T >::FunPtr) (double, double, double, int, int)
```

Definition at line 271 of file dpq.h.

### 6.530.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::Q2< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 270 of file dpq.h.

## 6.530.3 Constructor & Destructor Documentation

### 6.530.3.1 Q2()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::Q2< RTYPE, NA, T >::Q2 (
    FunPtr ptr_,
    const VEC_TYPE & vec_,
    double p0_,
    double p1_,
    bool lower_tail = true,
    bool log_ = false ) [inline]
```

Definition at line 273 of file dpq.h.

## 6.530.4 Member Function Documentation

### 6.530.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q2< RTYPE, NA, T >::operator[] (
    int i ) const [inline]
```

Definition at line 277 of file dpq.h.

References [Rcpp::stats::Q2< RTYPE, NA, T >::log](#), [Rcpp::stats::Q2< RTYPE, NA, T >::lower](#), [Rcpp::stats::Q2< RTYPE, NA, T >::p0](#), [Rcpp::stats::Q2< RTYPE, NA, T >::p1](#), [Rcpp::stats::Q2< RTYPE, NA, T >::ptr](#), and [Rcpp::stats::Q2< RTYPE, NA, T >::vec](#).

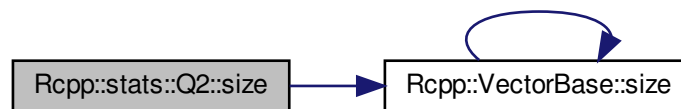
### 6.530.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q2< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 281 of file dpq.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::stats::Q2< RTYPE, NA, T >::vec.

Here is the call graph for this function:



## 6.530.5 Member Data Documentation

### 6.530.5.1 log

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q2< RTYPE, NA, T >::log [private]
```

Definition at line 287 of file dpq.h.

Referenced by Rcpp::stats::Q2< RTYPE, NA, T >::operator[]().

### 6.530.5.2 lower

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q2< RTYPE, NA, T >::lower [private]
```

Definition at line 287 of file dpq.h.

Referenced by Rcpp::stats::Q2< RTYPE, NA, T >::operator[]().

### 6.530.5.3 p0

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::Q2< RTYPE, NA, T >::p0 [private]
```

Definition at line 286 of file dpq.h.

Referenced by Rcpp::stats::Q2< RTYPE, NA, T >::operator[]().

### 6.530.5.4 p1

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::Q2< RTYPE, NA, T >::p1 [private]
```

Definition at line 286 of file dpq.h.

Referenced by Rcpp::stats::Q2< RTYPE, NA, T >::operator[]().

### 6.530.5.5 ptr

```
template<int RTYPE, bool NA, typename T >  
FunPtr Rcpp::stats::Q2< RTYPE, NA, T >::ptr [private]
```

Definition at line 284 of file dpq.h.

Referenced by Rcpp::stats::Q2< RTYPE, NA, T >::operator[]().

### 6.530.5.6 vec

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::stats::Q2< RTYPE, NA, T >::vec [private]
```

Definition at line 285 of file dpq.h.

Referenced by Rcpp::stats::Q2< RTYPE, NA, T >::operator[](), and Rcpp::stats::Q2< RTYPE, NA, T >::size().

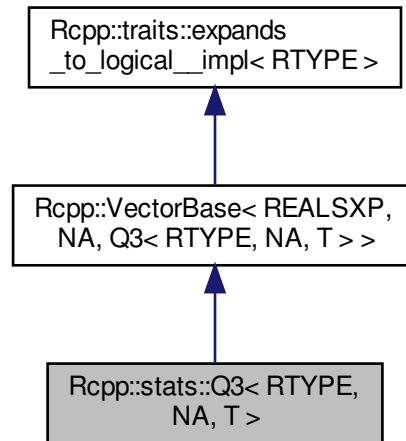
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/dpq/dpq.h](#)

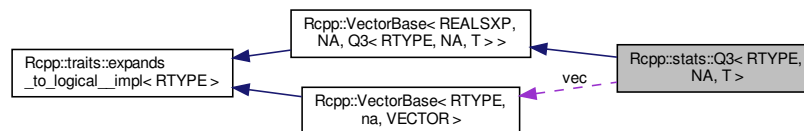
## 6.531 Rcpp::stats::Q3< RTYPE, NA, T > Class Template Reference

```
#include <dpq.h>
```

Inheritance diagram for Rcpp::stats::Q3< RTYPE, NA, T >:



Collaboration diagram for Rcpp::stats::Q3< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `double(* FunPtr)` (`double, double, double, double, int, int`)

### Public Member Functions

- `Q3` (`FunPtr ptr_`, `const VEC_TYPE &vec_`, `double p0_`, `double p1_`, `double p2_`, `bool lower_tail=true`, `bool log_↵=false`)
- `double operator[]` (`int i`) `const`
- `int size` () `const`



## Private Attributes

- [FunPtr ptr](#)
- const [VEC\\_TYPE](#) & [vec](#)
- double [p0](#)
- double [p1](#)
- double [p2](#)
- int [lower](#)
- int [log](#)

### 6.531.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::stats::Q3< RTYPE, NA, T >
```

Definition at line 292 of file dpq.h.

### 6.531.2 Member Typedef Documentation

#### 6.531.2.1 FunPtr

```
template<int RTYPE, bool NA, typename T >  
typedef double(* Rcpp::stats::Q3< RTYPE, NA, T >::FunPtr) (double, double, double, double, int,  
int)
```

Definition at line 295 of file dpq.h.

#### 6.531.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::stats::Q3< RTYPE, NA, T >::VEC\_TYPE
```

Definition at line 294 of file dpq.h.

### 6.531.3 Constructor & Destructor Documentation

### 6.531.3.1 Q3()

```
template<int RTYPE, bool NA, typename T >
Rcpp::stats::Q3< RTYPE, NA, T >::Q3 (
    FunPtr ptr_,
    const VEC_TYPE & vec_,
    double p0_,
    double p1_,
    double p2_,
    bool lower_tail = true,
    bool log_ = false ) [inline]
```

Definition at line 297 of file dpq.h.

## 6.531.4 Member Function Documentation

### 6.531.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q3< RTYPE, NA, T >::operator[] (
    int i ) const [inline]
```

Definition at line 301 of file dpq.h.

References `Rcpp::stats::Q3< RTYPE, NA, T >::log`, `Rcpp::stats::Q3< RTYPE, NA, T >::lower`, `Rcpp::stats::Q3< RTYPE, NA, T >::p0`, `Rcpp::stats::Q3< RTYPE, NA, T >::p1`, `Rcpp::stats::Q3< RTYPE, NA, T >::p2`, `Rcpp::stats::Q3< RTYPE, NA, T >::ptr`, and `Rcpp::stats::Q3< RTYPE, NA, T >::vec`.

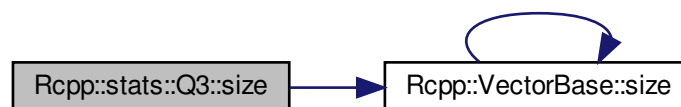
### 6.531.4.2 size()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::stats::Q3< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 305 of file dpq.h.

References `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`, and `Rcpp::stats::Q3< RTYPE, NA, T >::vec`.

Here is the call graph for this function:



## 6.531.5 Member Data Documentation

### 6.531.5.1 log

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::Q3< RTYPE, NA, T >::log [private]
```

Definition at line 311 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[]().

### 6.531.5.2 lower

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::stats::Q3< RTYPE, NA, T >::lower [private]
```

Definition at line 311 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[]().

### 6.531.5.3 p0

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::Q3< RTYPE, NA, T >::p0 [private]
```

Definition at line 310 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[]().

### 6.531.5.4 p1

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::stats::Q3< RTYPE, NA, T >::p1 [private]
```

Definition at line 310 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[]().

### 6.531.5.5 p2

```
template<int RTYPE, bool NA, typename T >
double Rcpp::stats::Q3< RTYPE, NA, T >::p2 [private]
```

Definition at line 310 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[]().

### 6.531.5.6 ptr

```
template<int RTYPE, bool NA, typename T >
FunPtr Rcpp::stats::Q3< RTYPE, NA, T >::ptr [private]
```

Definition at line 308 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[]().

### 6.531.5.7 vec

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::stats::Q3< RTYPE, NA, T >::vec [private]
```

Definition at line 309 of file dpq.h.

Referenced by Rcpp::stats::Q3< RTYPE, NA, T >::operator[](), and Rcpp::stats::Q3< RTYPE, NA, T >::size().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/dpq/dpq.h](#)

## 6.532 Rcpp::traits::expands\_to\_logical\_impl< LGLSXP >::r\_expands\_to\_logical Struct Reference

```
#include <expands_to_logical.h>
```

### 6.532.1 Detailed Description

Definition at line 37 of file expands\_to\_logical.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/expands\\_to\\_logical.h](#)

## 6.533 Rcpp::MatrixBase< RTYPE, na, MATRIX >::r\_matrix\_interface Struct Reference

```
#include <MatrixBase.h>
```

### 6.533.1 Detailed Description

```
template<int RTYPE, bool na, typename MATRIX>  
struct Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_matrix_interface
```

Definition at line 35 of file MatrixBase.h.

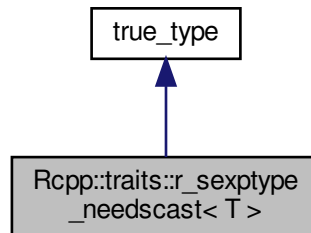
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/MatrixBase.h](#)

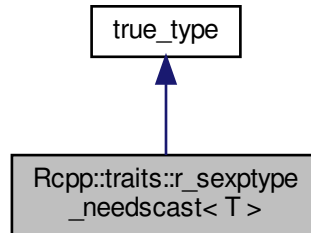
## 6.534 Rcpp::traits::r\_sexptype\_needscast< T > Struct Template Reference

```
#include <r_sexptype_traits.h>
```

Inheritance diagram for Rcpp::traits::r\_sexptype\_needscast< T >:



Collaboration diagram for `Rcpp::traits::r_sexptype_needscast< T >`:



## Additional Inherited Members

### 6.534.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::r_sexptype_needscast< T >
```

Indicates if a primitive type needs a `static_cast`

Definition at line 69 of file `r_sexptype_traits.h`.

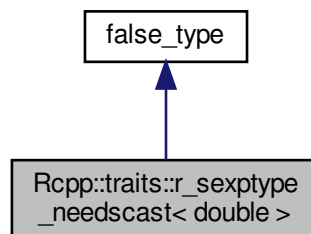
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

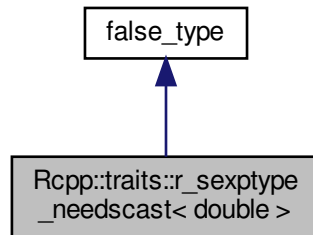
### 6.535 `Rcpp::traits::r_sexptype_needscast< double >` Struct Reference

```
#include <r_sexptype_traits.h>
```

Inheritance diagram for `Rcpp::traits::r_sexptype_needscast< double >`:



Collaboration diagram for Rcpp::traits::r\_sexptype\_needs cast< double >:



## Additional Inherited Members

### 6.535.1 Detailed Description

Definition at line 71 of file `r_sexptype_traits.h`.

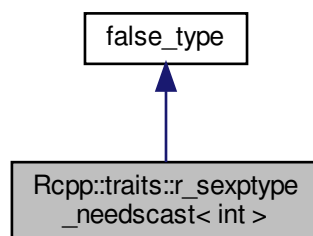
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

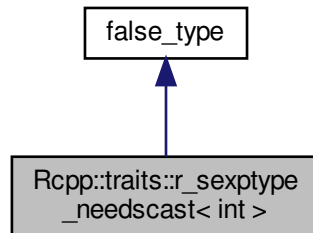
## 6.536 Rcpp::traits::r\_sexptype\_needs cast< int > Struct Reference

```
#include <r_sexptype_traits.h>
```

Inheritance diagram for Rcpp::traits::r\_sexptype\_needs cast< int >:



Collaboration diagram for `Rcpp::traits::r_sexptype_needs cast< int >`:



## Additional Inherited Members

### 6.536.1 Detailed Description

Definition at line 70 of file `r_sexptype_traits.h`.

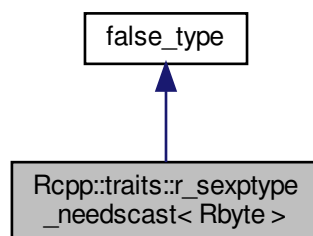
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

### 6.537 `Rcpp::traits::r_sexptype_needs cast< Rbyte >` Struct Reference

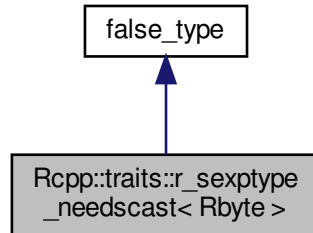
```
#include <r_sexptype_traits.h>
```

Inheritance diagram for `Rcpp::traits::r_sexptype_needs cast< Rbyte >`:





Collaboration diagram for Rcpp::traits::r\_sexptype\_needscast< Rbyte >:



## Additional Inherited Members

### 6.537.1 Detailed Description

Definition at line 73 of file r\_sexptype\_traits.h.

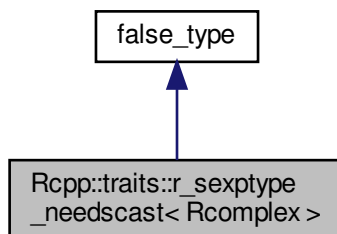
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

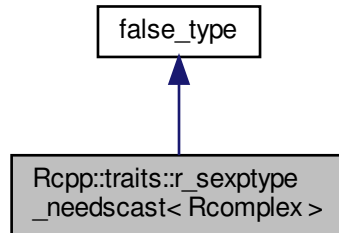
## 6.538 Rcpp::traits::r\_sexptype\_needscast< Rcomplex > Struct Reference

```
#include <r_sexptype_traits.h>
```

Inheritance diagram for Rcpp::traits::r\_sexptype\_needscast< Rcomplex >:



Collaboration diagram for `Rcpp::traits::r_sexptype_needs cast< Rcomplex >`:



## Additional Inherited Members

### 6.538.1 Detailed Description

Definition at line 72 of file `r_sexptype_traits.h`.

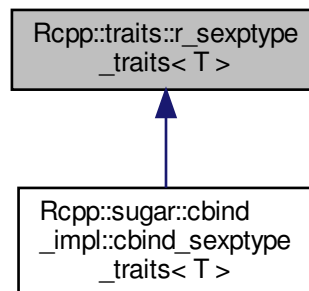
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

### 6.539 `Rcpp::traits::r_sexptype_traits< T >` Struct Template Reference

```
#include <r_sexptype_traits.h>
```

Inheritance diagram for `Rcpp::traits::r_sexptype_traits< T >`:



## Public Types

- enum { `rtype` = VECSEX }

### 6.539.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::r_sexptype_traits< T >
```

template that returns the SEXP type that is appropriate for the type T, this is allways VECSEX (lists) unless it is specialized

Definition at line 33 of file `r_sexptype_traits.h`.

### 6.539.2 Member Enumeration Documentation

#### 6.539.2.1 anonymous enum

```
template<typename T >
anonymous enum
```

##### Enumerator

rtype	
-------	--

Definition at line 33 of file `r_sexptype_traits.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.540 Rcpp::traits::r\_sexptype\_traits< bool > Struct Reference

```
#include <r_sexptype_traits.h>
```

## Public Types

- enum { `rtype` = LGLSEX }

### 6.540.1 Detailed Description

Definition at line 38 of file `r_sexptype_traits.h`.

### 6.540.2 Member Enumeration Documentation

#### 6.540.2.1 anonymous enum

anonymous enum

Enumerator

rtype	
-------	--

Definition at line 38 of file `r_sexptype_traits.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.541 Rcpp::traits::r\_sexptype\_traits< const double > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { [rtype](#) = REALSXP }

### 6.541.1 Detailed Description

Definition at line 37 of file `r_sexptype_traits.h`.

### 6.541.2 Member Enumeration Documentation

#### 6.541.2.1 anonymous enum

anonymous enum

Enumerator

rtype	
-------	--

Definition at line 37 of file r\_sexptype\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.542 Rcpp::traits::r\_sexptype\_traits< const int > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { [rtype](#) = INTSXP }

### 6.542.1 Detailed Description

Definition at line 35 of file r\_sexptype\_traits.h.

### 6.542.2 Member Enumeration Documentation

#### 6.542.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	
-------	--

Definition at line 35 of file r\_sexptype\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.543 Rcpp::traits::r\_sexptype\_traits< double > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { `rtype` = REALSXP }

### 6.543.1 Detailed Description

Definition at line 36 of file `r_sexptype_traits.h`.

### 6.543.2 Member Enumeration Documentation

#### 6.543.2.1 anonymous enum

anonymous enum

Enumerator

rtype	
-------	--

Definition at line 36 of file `r_sexptype_traits.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.544 Rcpp::traits::r\_sexptype\_traits< float > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { `rtype` = REALSXP }

### 6.544.1 Detailed Description

Definition at line 45 of file r\_sexptype\_traits.h.

### 6.544.2 Member Enumeration Documentation

#### 6.544.2.1 anonymous enum

anonymous enum

Enumerator

rtype	
-------	--

Definition at line 45 of file r\_sexptype\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.545 Rcpp::traits::r\_sexptype\_traits< int > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { [rtype](#) = INTSXP }

### 6.545.1 Detailed Description

Definition at line 34 of file r\_sexptype\_traits.h.

### 6.545.2 Member Enumeration Documentation

#### 6.545.2.1 anonymous enum

anonymous enum

**Enumerator**

rtype	
-------	--

Definition at line 34 of file r\_sexptype\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.546 Rcpp::traits::r\_sexptype\_traits< long > Struct Reference

```
#include <r_sexptype_traits.h>
```

**Public Types**

- enum { [rtype](#) = REALSXP }

### 6.546.1 Detailed Description

Definition at line 49 of file r\_sexptype\_traits.h.

### 6.546.2 Member Enumeration Documentation

#### 6.546.2.1 anonymous enum

```
anonymous enum
```

**Enumerator**

rtype	
-------	--

Definition at line 49 of file r\_sexptype\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)



## 6.547 Rcpp::traits::r\_sexptype\_traits< long double > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { `rtype` = REALSXP }

### 6.547.1 Detailed Description

Definition at line 55 of file `r_sexptype_traits.h`.

### 6.547.2 Member Enumeration Documentation

#### 6.547.2.1 anonymous enum

anonymous enum

Enumerator

rtype	
-------	--

Definition at line 55 of file `r_sexptype_traits.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.548 Rcpp::traits::r\_sexptype\_traits< Rbyte > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { `rtype` = RAWXP }

### 6.548.1 Detailed Description

Definition at line 41 of file `r_sexptype_traits.h`.

### 6.548.2 Member Enumeration Documentation

#### 6.548.2.1 anonymous enum

anonymous enum

Enumerator

rtype	
-------	--

Definition at line 41 of file `r_sexptype_traits.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.549 Rcpp::traits::r\_sexptype\_traits< Rcomplex > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { [rtype](#) = CPLXSP }

### 6.549.1 Detailed Description

Definition at line 40 of file `r_sexptype_traits.h`.

### 6.549.2 Member Enumeration Documentation

#### 6.549.2.1 anonymous enum

anonymous enum

Enumerator

rtype	
-------	--

Definition at line 40 of file r\_sexptype\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.550 Rcpp::traits::r\_sexptype\_traits< Rcpp::Date > Struct Reference

```
#include <Date.h>
```

### Public Types

- enum { [rtype](#) = REALSXP }

### 6.550.1 Detailed Description

Definition at line 37 of file Date.h.

### 6.550.2 Member Enumeration Documentation

#### 6.550.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	
-------	--

Definition at line 38 of file Date.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/api/bones/Date.h](#)

## 6.551 Rcpp::traits::r\_sexptype\_traits< Rcpp::Datetime > Struct Reference

```
#include <Datetime.h>
```

### Public Types

- enum { `rtype` = REALSXP }

### 6.551.1 Detailed Description

Definition at line 37 of file `Datetime.h`.

### 6.551.2 Member Enumeration Documentation

#### 6.551.2.1 anonymous enum

anonymous enum

Enumerator

rtype	
-------	--

Definition at line 38 of file `Datetime.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/api/bones/Datetime.h`

## 6.552 Rcpp::traits::r\_sexptype\_traits< Rcpp::String > Struct Reference

```
#include <String.h>
```

### Public Types

- enum { `rtype` = STRSXP }

### 6.552.1 Detailed Description

Definition at line 677 of file String.h.

### 6.552.2 Member Enumeration Documentation

#### 6.552.2.1 anonymous enum

anonymous enum

Enumerator

rtype	
-------	--

Definition at line 677 of file String.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/[String.h](#)

## 6.553 Rcpp::traits::r\_sexptype\_traits< short > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { [rtype](#) = INTSXP }

### 6.553.1 Detailed Description

Definition at line 59 of file r\_sexptype\_traits.h.

### 6.553.2 Member Enumeration Documentation

#### 6.553.2.1 anonymous enum

anonymous enum

**Enumerator**

rtype	
-------	--

Definition at line 59 of file r\_sexptype\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.554 Rcpp::traits::r\_sexptype\_traits< std::complex< double > > Struct Reference

```
#include <r_sexptype_traits.h>
```

**Public Types**

- enum { [rtype](#) = CPLXSPX }

### 6.554.1 Detailed Description

Definition at line 63 of file r\_sexptype\_traits.h.

### 6.554.2 Member Enumeration Documentation

#### 6.554.2.1 anonymous enum

```
anonymous enum
```

**Enumerator**

rtype	
-------	--

Definition at line 63 of file r\_sexptype\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.555 Rcpp::traits::r\_sexptype\_traits< std::complex< float > > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { `rtype` = CPLXSP }

### 6.555.1 Detailed Description

Definition at line 64 of file `r_sexptype_traits.h`.

### 6.555.2 Member Enumeration Documentation

#### 6.555.2.1 anonymous enum

anonymous enum

#### Enumerator

<code>rtype</code>	
--------------------	--

Definition at line 64 of file `r_sexptype_traits.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/r_sexptype_traits.h`

## 6.556 Rcpp::traits::r\_sexptype\_traits< std::string > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { `rtype` = STRSP }

### 6.556.1 Detailed Description

Definition at line 39 of file `r_sexptype_traits.h`.

### 6.556.2 Member Enumeration Documentation

#### 6.556.2.1 anonymous enum

anonymous enum

Enumerator

rtype	
-------	--

Definition at line 39 of file `r_sexptype_traits.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.557 Rcpp::traits::r\_sexptype\_traits< unsigned int > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { [rtype](#) = REALSXP }

### 6.557.1 Detailed Description

Definition at line 44 of file `r_sexptype_traits.h`.

### 6.557.2 Member Enumeration Documentation

#### 6.557.2.1 anonymous enum

anonymous enum



Enumerator

rtype	
-------	--

Definition at line 44 of file r\_sexptype\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.558 Rcpp::traits::r\_sexptype\_traits< unsigned long > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { [rtype](#) = REALSXP }

### 6.558.1 Detailed Description

Definition at line 50 of file r\_sexptype\_traits.h.

### 6.558.2 Member Enumeration Documentation

#### 6.558.2.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	
-------	--

Definition at line 50 of file r\_sexptype\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

## 6.559 Rcpp::traits::r\_sexptype\_traits< unsigned short > Struct Reference

```
#include <r_sexptype_traits.h>
```

### Public Types

- enum { `rtype` = INTSXP }

### 6.559.1 Detailed Description

Definition at line 60 of file `r_sexptype_traits.h`.

### 6.559.2 Member Enumeration Documentation

#### 6.559.2.1 anonymous enum

anonymous enum

Enumerator

rtype	
-------	--

Definition at line 60 of file `r_sexptype_traits.h`.

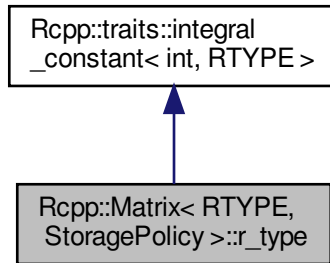
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_sexptype\\_traits.h](#)

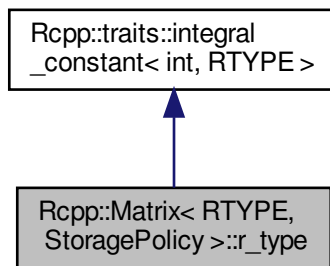
## 6.560 Rcpp::Matrix< RTYPE, StoragePolicy >::r\_type Struct Reference

```
#include <Matrix.h>
```

Inheritance diagram for Rcpp::Matrix< RTYPE, StoragePolicy >::r\_type:



Collaboration diagram for Rcpp::Matrix< RTYPE, StoragePolicy >::r\_type:



## Additional Inherited Members

### 6.560.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>  
struct Rcpp::Matrix< RTYPE, StoragePolicy >::r_type
```

Definition at line 34 of file Matrix.h.

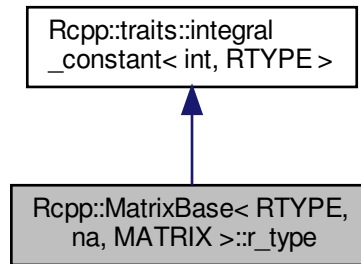
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/[Matrix.h](#)

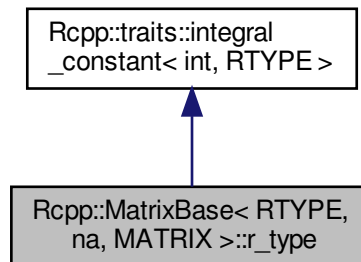
## 6.561 Rcpp::MatrixBase< RTYPE, na, MATRIX >::r\_type Struct Reference

```
#include <MatrixBase.h>
```

Inheritance diagram for Rcpp::MatrixBase< RTYPE, na, MATRIX >::r\_type:



Collaboration diagram for Rcpp::MatrixBase< RTYPE, na, MATRIX >::r\_type:



### Additional Inherited Members

#### 6.561.1 Detailed Description

```
template<int RTYPE, bool na, typename MATRIX>
struct Rcpp::MatrixBase< RTYPE, na, MATRIX >::r_type
```

Definition at line 34 of file MatrixBase.h.

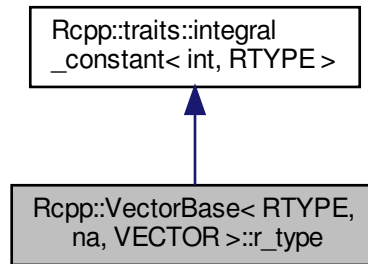
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/[MatrixBase.h](#)

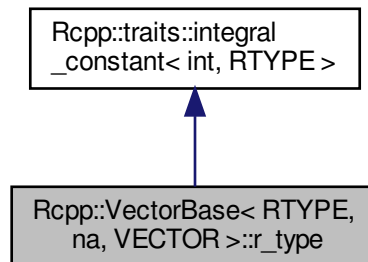
## 6.562 Rcpp::VectorBase< RTYPE, na, VECTOR >::r\_type Struct Reference

```
#include <VectorBase.h>
```

Inheritance diagram for Rcpp::VectorBase< RTYPE, na, VECTOR >::r\_type:



Collaboration diagram for Rcpp::VectorBase< RTYPE, na, VECTOR >::r\_type:



### Additional Inherited Members

#### 6.562.1 Detailed Description

```
template<int RTYPE, bool na, typename VECTOR>
struct Rcpp::VectorBase< RTYPE, na, VECTOR >::r_type
```

Definition at line 32 of file VectorBase.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/[VectorBase.h](#)

## 6.563 Rcpp::traits::r\_type\_enum\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.563.1 Detailed Description

identifies an enum. conversions from/to int is used

Definition at line 97 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.564 Rcpp::traits::r\_type\_generic\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.564.1 Detailed Description

Default

Definition at line 44 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.565 Rcpp::traits::r\_type\_module\_object\_const\_pointer\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.565.1 Detailed Description

identifies a module object const pointer

Definition at line 76 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.566 Rcpp::traits::r\_type\_module\_object\_const\_reference\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.566.1 Detailed Description

identifies a const reference to a module object.

Definition at line 92 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.567 Rcpp::traits::r\_type\_module\_object\_pointer\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.567.1 Detailed Description

identifies a module object pointer

Definition at line 71 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.568 Rcpp::traits::r\_type\_module\_object\_reference\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.568.1 Detailed Description

identifies a reference to a module object.

Definition at line 87 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.569 Rcpp::traits::r\_type\_module\_object\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.569.1 Detailed Description

identifies a module object. Implementers of modules can define the [r\\_type\\_traits](#) to show that their object is handled

Definition at line 82 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.570 Rcpp::traits::r\_type\_pair\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.570.1 Detailed Description

Identifies a pair<const KEY, VALUE>, used to wrap map<KEY, VALUE> VALUE can be anything that wraps, KEY can be anything we can build a [String](#) from

Definition at line 66 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.571 Rcpp::traits::r\_type\_pairstring\_generic\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.571.1 Detailed Description

Identifies pair<const std::string,T>

Definition at line 60 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)



## 6.572 Rcpp::traits::r\_type\_pairstring\_primitive\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.572.1 Detailed Description

Identifies that the type if pair<const std::string,T> where T is a primitive type

Definition at line 50 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.573 Rcpp::traits::r\_type\_pairstring\_string\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.573.1 Detailed Description

Identifies that the associated type is pair<const std::(w)?string,std::(w)?string>

Definition at line 55 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.574 Rcpp::traits::r\_type\_primitive\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.574.1 Detailed Description

Identifies a primitive type that needs no special handling int, double, Rbyte, Rcomplex

Definition at line 33 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.575 Rcpp::traits::r\_type\_RcppString\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.575.1 Detailed Description

Definition at line 99 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.576 Rcpp::traits::r\_type\_string\_tag Struct Reference

```
#include <r_type_traits.h>
```

### 6.576.1 Detailed Description

Identifies that the associated type can be implicitly converted to std::string

Definition at line 39 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.577 Rcpp::traits::r\_type\_traits< T > Struct Template Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_generic\\_tag](#) r\_category

### 6.577.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::r_type_traits< T >
```

[R](#) type trait. Helps wrap.

Definition at line 104 of file r\_type\_traits.h.

## 6.577.2 Member Typedef Documentation

### 6.577.2.1 r\_category

```
template<typename T >
typedef r\_type\_generic\_tag Rcpp::traits::r_type_traits< T >::r_category
```

Definition at line 104 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.578 Rcpp::traits::r\_type\_traits< bool > Struct Reference

```
#include <r\_type\_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) r\_category

### 6.578.1 Detailed Description

Definition at line 141 of file [r\\_type\\_traits.h](#).

## 6.578.2 Member Typedef Documentation

### 6.578.2.1 r\_category

```
typedef r\_type\_primitive\_tag Rcpp::traits::r_type_traits< bool >::r_category
```

Definition at line 141 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.579 Rcpp::traits::r\_type\_traits< char > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_string\\_tag](#) [r\\_category](#)

### 6.579.1 Detailed Description

Definition at line 144 of file [r\\_type\\_traits.h](#).

### 6.579.2 Member Typedef Documentation

#### 6.579.2.1 r\_category

```
typedef r\_type\_string\_tag Rcpp::traits::r_type_traits< char >::r_category
```

Definition at line 144 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.580 Rcpp::traits::r\_type\_traits< const char \* > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_string\\_tag](#) [r\\_category](#)

### 6.580.1 Detailed Description

Definition at line 149 of file [r\\_type\\_traits.h](#).

## 6.580.2 Member Typedef Documentation

### 6.580.2.1 r\_category

```
typedef r_type_string_tag Rcpp::traits::r_type_traits< const char * >::r_category
```

Definition at line 149 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.581 Rcpp::traits::r\_type\_traits< const double > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) r\_category

### 6.581.1 Detailed Description

Definition at line 138 of file r\_type\_traits.h.

## 6.581.2 Member Typedef Documentation

### 6.581.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< const double >::r_category
```

Definition at line 138 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.582 Rcpp::traits::r\_type\_traits< const int > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

### 6.582.1 Detailed Description

Definition at line 136 of file [r\\_type\\_traits.h](#).

### 6.582.2 Member Typedef Documentation

#### 6.582.2.1 r\_category

```
typedef r\_type\_primitive\_tag Rcpp::traits::r_type_traits< const int >::r_category
```

Definition at line 136 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.583 Rcpp::traits::r\_type\_traits< const wchar\_t \* > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_string\\_tag](#) [r\\_category](#)

### 6.583.1 Detailed Description

Definition at line 150 of file [r\\_type\\_traits.h](#).

## 6.583.2 Member Typedef Documentation

### 6.583.2.1 r\_category

```
typedef r_type_string_tag Rcpp::traits::r_type_traits< const wchar_t * >::r_category
```

Definition at line 150 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.584 Rcpp::traits::r\_type\_traits< double > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) r\_category

### 6.584.1 Detailed Description

Definition at line 137 of file r\_type\_traits.h.

## 6.584.2 Member Typedef Documentation

### 6.584.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< double >::r_category
```

Definition at line 137 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.585 Rcpp::traits::r\_type\_traits< float > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

### 6.585.1 Detailed Description

Definition at line 148 of file [r\\_type\\_traits.h](#).

### 6.585.2 Member Typedef Documentation

#### 6.585.2.1 r\_category

```
typedef r\_type\_primitive\_tag Rcpp::traits::r_type_traits< float >::r_category
```

Definition at line 148 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.586 Rcpp::traits::r\_type\_traits< int > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

### 6.586.1 Detailed Description

Definition at line 135 of file [r\\_type\\_traits.h](#).



## 6.586.2 Member Typedef Documentation

### 6.586.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< int >::r_category
```

Definition at line 135 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.587 Rcpp::traits::r\_type\_traits< long > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) r\_category

### 6.587.1 Detailed Description

Definition at line 153 of file r\_type\_traits.h.

## 6.587.2 Member Typedef Documentation

### 6.587.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< long >::r_category
```

Definition at line 153 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.588 Rcpp::traits::r\_type\_traits< long double > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

### 6.588.1 Detailed Description

Definition at line 161 of file [r\\_type\\_traits.h](#).

### 6.588.2 Member Typedef Documentation

#### 6.588.2.1 r\_category

```
typedef r\_type\_primitive\_tag Rcpp::traits::r_type_traits< long double >::r_category
```

Definition at line 161 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.589 Rcpp::traits::r\_type\_traits< Rbyte > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

### 6.589.1 Detailed Description

Definition at line 139 of file [r\\_type\\_traits.h](#).

## 6.589.2 Member Typedef Documentation

### 6.589.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< Rbyte >::r_category
```

Definition at line 139 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.590 Rcpp::traits::r\_type\_traits< Rcomplex > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag r\\_category](#)

### 6.590.1 Detailed Description

Definition at line 140 of file r\_type\_traits.h.

## 6.590.2 Member Typedef Documentation

### 6.590.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< Rcomplex >::r_category
```

Definition at line 140 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.591 Rcpp::traits::r\_type\_traits< Rcpp::Date > Struct Reference

```
#include <Date.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

#### 6.591.1 Detailed Description

Definition at line 31 of file Date.h.

#### 6.591.2 Member Typedef Documentation

##### 6.591.2.1 r\_category

```
typedef r\_type\_primitive\_tag Rcpp::traits::r\_type\_traits< Rcpp::Date >::r_category
```

Definition at line 32 of file Date.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/api/bones/Date.h](#)

## 6.592 Rcpp::traits::r\_type\_traits< Rcpp::Datetime > Struct Reference

```
#include <Datetime.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

#### 6.592.1 Detailed Description

Definition at line 31 of file Datetime.h.

## 6.592.2 Member Typedef Documentation

### 6.592.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< Rcpp::Datetime >::r_category
```

Definition at line 32 of file Datetime.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/bones/Datetime.h

## 6.593 Rcpp::traits::r\_type\_traits< Rcpp::object< T > > Struct Template Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef r\_type\_module\_object\_pointer\_tag r\_category

### 6.593.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::r_type_traits< Rcpp::object< T > >
```

module object type

Definition at line 109 of file r\_type\_traits.h.

### 6.593.2 Member Typedef Documentation

### 6.593.2.1 `r_category`

```
template<typename T >
typedef r\_type\_module\_object\_pointer\_tag Rcpp::traits::r_type_traits< Rcpp::object< T > >←
::r_category
```

Definition at line 109 of file `r_type_traits.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.594 `Rcpp::traits::r_type_traits< Rcpp::String >` Struct Reference

```
#include <String.h>
```

### Public Types

- typedef [r\\_type\\_RcppString\\_tag](#) `r_category`

### 6.594.1 Detailed Description

Definition at line 676 of file `String.h`.

### 6.594.2 Member Typedef Documentation

#### 6.594.2.1 `r_category`

```
typedef r\_type\_RcppString\_tag Rcpp::traits::r_type_traits< Rcpp::String >::r_category
```

Definition at line 676 of file `String.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/String.h](#)

## 6.595 `Rcpp::traits::r_type_traits< short >` Struct Reference

```
#include <r_type_traits.h>
```

## Public Types

- [typedef r\\_type\\_primitive\\_tag r\\_category](#)

### 6.595.1 Detailed Description

Definition at line 165 of file r\_type\_traits.h.

### 6.595.2 Member Typedef Documentation

#### 6.595.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< short >::r_category
```

Definition at line 165 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.596 Rcpp::traits::r\_type\_traits< std::complex< double > > Struct Reference

```
#include <r_type_traits.h>
```

## Public Types

- [typedef r\\_type\\_primitive\\_tag r\\_category](#)

### 6.596.1 Detailed Description

Definition at line 173 of file r\_type\_traits.h.

### 6.596.2 Member Typedef Documentation

### 6.596.2.1 `r_category`

```
typedef r\_type\_primitive\_tag Rcpp::traits::r_type_traits< std::complex< double > >::r_category
```

Definition at line 173 of file `r_type_traits.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.597 `Rcpp::traits::r_type_traits< std::complex< float > >` Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) `r_category`

### 6.597.1 Detailed Description

Definition at line 177 of file `r_type_traits.h`.

### 6.597.2 Member Typedef Documentation

#### 6.597.2.1 `r_category`

```
typedef r\_type\_primitive\_tag Rcpp::traits::r_type_traits< std::complex< float > >::r_category
```

Definition at line 177 of file `r_type_traits.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.598 `Rcpp::traits::r_type_traits< std::pair< const KEY, VALUE > >` Struct Template Reference

```
#include <r_type_traits.h>
```



## Public Types

- [typedef r\\_type\\_pair\\_tag r\\_category](#)

### 6.598.1 Detailed Description

```
template<typename KEY, typename VALUE>
struct Rcpp::traits::r_type_traits< std::pair< const KEY, VALUE > >
```

Definition at line 113 of file r\_type\_traits.h.

### 6.598.2 Member Typedef Documentation

#### 6.598.2.1 r\_category

```
template<typename KEY , typename VALUE >
typedef r\_type\_pair\_tag Rcpp::traits::r_type_traits< std::pair< const KEY, VALUE > >::r_category
```

Definition at line 114 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.599 Rcpp::traits::r\_type\_traits< std::pair< const std::string, bool > > Struct Reference

```
#include <r_type_traits.h>
```

## Public Types

- [typedef r\\_type\\_pairstring\\_primitive\\_tag r\\_category](#)

### 6.599.1 Detailed Description

Definition at line 126 of file r\_type\_traits.h.

## 6.599.2 Member Typedef Documentation

### 6.599.2.1 r\_category

```
typedef r_type_pairstring_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string,  
bool > >::r_category
```

Definition at line 126 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.600 Rcpp::traits::r\_type\_traits< std::pair< const std::string, char > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_pairstring\\_string\\_tag](#) r\_category

### 6.600.1 Detailed Description

Definition at line 129 of file r\_type\_traits.h.

## 6.600.2 Member Typedef Documentation

### 6.600.2.1 r\_category

```
typedef r_type_pairstring_string_tag Rcpp::traits::r_type_traits< std::pair< const std::string,  
char > >::r_category
```

Definition at line 129 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.601 Rcpp::traits::r\_type\_traits< std::pair< const std::string, const int > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_pairstring\\_primitive\\_tag](#) [r\\_category](#)

#### 6.601.1 Detailed Description

Definition at line 122 of file [r\\_type\\_traits.h](#).

#### 6.601.2 Member Typedef Documentation

##### 6.601.2.1 [r\\_category](#)

```
typedef r\_type\_pairstring\_primitive\_tag Rcpp::traits::r_type_traits< std::pair< const std::string,  
const int > >::r_category
```

Definition at line 122 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.602 Rcpp::traits::r\_type\_traits< std::pair< const std::string, double > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_pairstring\\_primitive\\_tag](#) [r\\_category](#)

#### 6.602.1 Detailed Description

Definition at line 123 of file [r\\_type\\_traits.h](#).

## 6.602.2 Member Typedef Documentation

### 6.602.2.1 r\_category

```
typedef r_type_pairstring_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string,  
double > >::r_category
```

Definition at line 123 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.603 Rcpp::traits::r\_type\_traits< std::pair< const std::string, float > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_pairstring\\_primitive\\_tag r\\_category](#)

### 6.603.1 Detailed Description

Definition at line 133 of file r\_type\_traits.h.

## 6.603.2 Member Typedef Documentation

### 6.603.2.1 r\_category

```
typedef r_type_pairstring_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string,  
float > >::r_category
```

Definition at line 133 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.604 Rcpp::traits::r\_type\_traits< std::pair< const std::string, int > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_pairstring\\_primitive\\_tag](#) [r\\_category](#)

#### 6.604.1 Detailed Description

Definition at line 121 of file [r\\_type\\_traits.h](#).

#### 6.604.2 Member Typedef Documentation

##### 6.604.2.1 [r\\_category](#)

```
typedef r\_type\_pairstring\_primitive\_tag Rcpp::traits::r_type_traits< std::pair< const std::string,  
int > >::r_category
```

Definition at line 121 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.605 Rcpp::traits::r\_type\_traits< std::pair< const std::string, long > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

#### 6.605.1 Detailed Description

Definition at line 154 of file [r\\_type\\_traits.h](#).

## 6.605.2 Member Typedef Documentation

### 6.605.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, long >  
>::r_category
```

Definition at line 154 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.606 Rcpp::traits::r\_type\_traits< std::pair< const std::string, long double > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag r\\_category](#)

### 6.606.1 Detailed Description

Definition at line 162 of file r\_type\_traits.h.

## 6.606.2 Member Typedef Documentation

### 6.606.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, long  
double > >::r_category
```

Definition at line 162 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.607 Rcpp::traits::r\_type\_traits< std::pair< const std::string, Rbyte > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_pairstring\\_primitive\\_tag](#) [r\\_category](#)

#### 6.607.1 Detailed Description

Definition at line 124 of file [r\\_type\\_traits.h](#).

#### 6.607.2 Member Typedef Documentation

##### 6.607.2.1 [r\\_category](#)

```
typedef r\_type\_pairstring\_primitive\_tag Rcpp::traits::r_type_traits< std::pair< const std::string,  
Rbyte > >::r_category
```

Definition at line 124 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.608 Rcpp::traits::r\_type\_traits< std::pair< const std::string, Rcomplex > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_pairstring\\_primitive\\_tag](#) [r\\_category](#)

#### 6.608.1 Detailed Description

Definition at line 125 of file [r\\_type\\_traits.h](#).

## 6.608.2 Member Typedef Documentation

### 6.608.2.1 r\_category

```
typedef r_type_pairstring_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, Rcomplex > >::r_category
```

Definition at line 125 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.609 Rcpp::traits::r\_type\_traits< std::pair< const std::string, Rcpp::Date > > Struct Reference

```
#include <Date.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) r\_category

### 6.609.1 Detailed Description

Definition at line 34 of file Date.h.

## 6.609.2 Member Typedef Documentation

### 6.609.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Date > >::r_category
```

Definition at line 35 of file Date.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/api/bones/Date.h](#)



## 6.610 Rcpp::traits::r\_type\_traits< std::pair< const std::string, Rcpp::Datetime > > Struct Reference

```
#include <Datetime.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

#### 6.610.1 Detailed Description

Definition at line 34 of file [Datetime.h](#).

#### 6.610.2 Member Typedef Documentation

##### 6.610.2.1 [r\\_category](#)

```
typedef r\_type\_primitive\_tag Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Datetime > >::r_category
```

Definition at line 35 of file [Datetime.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/api/bones/Datetime.h](#)

## 6.611 Rcpp::traits::r\_type\_traits< std::pair< const std::string, short > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

#### 6.611.1 Detailed Description

Definition at line 166 of file [r\\_type\\_traits.h](#).

## 6.611.2 Member Typedef Documentation

### 6.611.2.1 `r_category`

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, short >  
>::r_category
```

Definition at line 166 of file `r_type_traits.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/r_type_traits.h`

## 6.612 `Rcpp::traits::r_type_traits`< `std::pair`< `const std::string`, `std::complex`< `double` > > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef `r_type_primitive_tag` `r_category`

### 6.612.1 Detailed Description

Definition at line 174 of file `r_type_traits.h`.

## 6.612.2 Member Typedef Documentation

### 6.612.2.1 `r_category`

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex<  
double > > >::r_category
```

Definition at line 174 of file `r_type_traits.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/r_type_traits.h`

## 6.613 Rcpp::traits::r\_type\_traits< std::pair< const std::string, std::complex< float > > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

#### 6.613.1 Detailed Description

Definition at line 178 of file `r_type_traits.h`.

#### 6.613.2 Member Typedef Documentation

##### 6.613.2.1 `r_category`

```
typedef r\_type\_primitive\_tag Rcpp::traits::r_type_traits< std::pair< const std::string, std::complex< float > > >::r_category
```

Definition at line 178 of file `r_type_traits.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/r_type_traits.h`

## 6.614 Rcpp::traits::r\_type\_traits< std::pair< const std::string, std::string > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_pairstring\\_string\\_tag](#) [r\\_category](#)

#### 6.614.1 Detailed Description

Definition at line 127 of file `r_type_traits.h`.

## 6.614.2 Member Typedef Documentation

### 6.614.2.1 r\_category

```
typedef r_type_pairstring_string_tag Rcpp::traits::r_type_traits< std::pair< const std::string,  
std::string > >::r_category
```

Definition at line 127 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.615 Rcpp::traits::r\_type\_traits< std::pair< const std::string, std::wstring > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_pairstring\\_string\\_tag](#) r\_category

### 6.615.1 Detailed Description

Definition at line 128 of file r\_type\_traits.h.

## 6.615.2 Member Typedef Documentation

### 6.615.2.1 r\_category

```
typedef r_type_pairstring_string_tag Rcpp::traits::r_type_traits< std::pair< const std::string,  
std::wstring > >::r_category
```

Definition at line 128 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.616 Rcpp::traits::r\_type\_traits< std::pair< const std::string, T > > Struct Template Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_pairstring\\_generic\\_tag](#) [r\\_category](#)

#### 6.616.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::r_type_traits< std::pair< const std::string, T > >
```

special cases pair<string,T> to deal with map<string,T> etc ...

Definition at line 120 of file r\_type\_traits.h.

#### 6.616.2 Member Typedef Documentation

##### 6.616.2.1 r\_category

```
template<typename T >  
typedef r\_type\_pairstring\_generic\_tag Rcpp::traits::r_type_traits< std::pair< const std::string,  
T > >::r_category
```

Definition at line 120 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[r\\_type\\_traits.h](#)

## 6.617 Rcpp::traits::r\_type\_traits< std::pair< const std::string, unsigned int > > Struct Reference

```
#include <r_type_traits.h>
```

## Public Types

- typedef [r\\_type\\_pairstring\\_primitive\\_tag](#) [r\\_category](#)

### 6.617.1 Detailed Description

Definition at line 132 of file [r\\_type\\_traits.h](#).

### 6.617.2 Member Typedef Documentation

#### 6.617.2.1 [r\\_category](#)

```
typedef r\_type\_pairstring\_primitive\_tag Rcpp::traits::r_type_traits< std::pair< const std::string,  
unsigned int > >::r_category
```

Definition at line 132 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.618 [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, unsigned long > >](#) Struct Reference

```
#include <r\_type\_traits.h>
```

## Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

### 6.618.1 Detailed Description

Definition at line 158 of file [r\\_type\\_traits.h](#).

### 6.618.2 Member Typedef Documentation

### 6.618.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned  
long > >::r_category
```

Definition at line 158 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.619 Rcpp::traits::r\_type\_traits< std::pair< const std::string, unsigned short > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) r\_category

### 6.619.1 Detailed Description

Definition at line 170 of file r\_type\_traits.h.

### 6.619.2 Member Typedef Documentation

#### 6.619.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< std::pair< const std::string, unsigned  
short > >::r_category
```

Definition at line 170 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.620 Rcpp::traits::r\_type\_traits< std::pair< const std::string, wchar\_t > > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_pairstring\\_string\\_tag](#) r\_category

#### 6.620.1 Detailed Description

Definition at line 130 of file `r_type_traits.h`.

#### 6.620.2 Member Typedef Documentation

##### 6.620.2.1 r\_category

```
typedef r\_type\_pairstring\_string\_tag Rcpp::traits::r_type_traits< std::pair< const std::string,  
wchar_t > >::r_category
```

Definition at line 130 of file `r_type_traits.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/r_type_traits.h`

## 6.621 Rcpp::traits::r\_type\_traits< std::string > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_string\\_tag](#) r\_category

#### 6.621.1 Detailed Description

Definition at line 142 of file `r_type_traits.h`.



## 6.621.2 Member Typedef Documentation

### 6.621.2.1 r\_category

```
typedef r_type_string_tag Rcpp::traits::r_type_traits< std::string >::r_category
```

Definition at line 142 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.622 Rcpp::traits::r\_type\_traits< std::wstring > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_string\\_tag](#) r\_category

### 6.622.1 Detailed Description

Definition at line 143 of file r\_type\_traits.h.

## 6.622.2 Member Typedef Documentation

### 6.622.2.1 r\_category

```
typedef r_type_string_tag Rcpp::traits::r_type_traits< std::wstring >::r_category
```

Definition at line 143 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.623 Rcpp::traits::r\_type\_traits< unsigned int > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

### 6.623.1 Detailed Description

Definition at line 147 of file [r\\_type\\_traits.h](#).

### 6.623.2 Member Typedef Documentation

#### 6.623.2.1 r\_category

```
typedef r\_type\_primitive\_tag Rcpp::traits::r_type_traits< unsigned int >::r_category
```

Definition at line 147 of file [r\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.624 Rcpp::traits::r\_type\_traits< unsigned long > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) [r\\_category](#)

### 6.624.1 Detailed Description

Definition at line 157 of file [r\\_type\\_traits.h](#).

## 6.624.2 Member Typedef Documentation

### 6.624.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< unsigned long >::r_category
```

Definition at line 157 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.625 Rcpp::traits::r\_type\_traits< unsigned short > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_primitive\\_tag](#) r\_category

### 6.625.1 Detailed Description

Definition at line 169 of file r\_type\_traits.h.

## 6.625.2 Member Typedef Documentation

### 6.625.2.1 r\_category

```
typedef r_type_primitive_tag Rcpp::traits::r_type_traits< unsigned short >::r_category
```

Definition at line 169 of file r\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.626 Rcpp::traits::r\_type\_traits< wchar\_t > Struct Reference

```
#include <r_type_traits.h>
```

### Public Types

- typedef [r\\_type\\_string\\_tag](#) [r\\_category](#)

### 6.626.1 Detailed Description

Definition at line 145 of file [r\\_type\\_traits.h](#).

### 6.626.2 Member Typedef Documentation

#### 6.626.2.1 r\_category

```
typedef r\_type\_string\_tag Rcpp::traits::r_type_traits< wchar_t >::r_category
```

Definition at line 145 of file [r\\_type\\_traits.h](#).

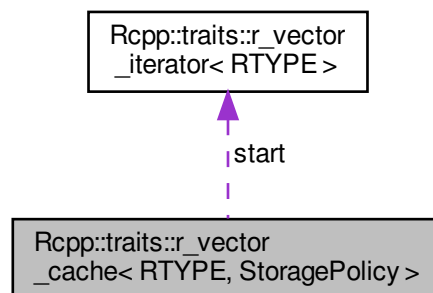
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/r\\_type\\_traits.h](#)

## 6.627 Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy > Class Template Reference

```
#include <traits.h>
```

Collaboration diagram for [Rcpp::traits::r\\_vector\\_cache< RTYPE, StoragePolicy >](#):



## Public Types

- typedef `Rcpp::Vector`< RTYPE, StoragePolicy > `VECTOR`
- typedef `r_vector_iterator`< RTYPE >::type `iterator`
- typedef `r_vector_const_iterator`< RTYPE >::type `const_iterator`
- typedef `r_vector_proxy`< RTYPE >::type `proxy`
- typedef `r_vector_const_proxy`< RTYPE >::type `const_proxy`
- typedef `storage_type`< RTYPE >::type `storage_type`

## Public Member Functions

- `r_vector_cache` ()
- void `update` (const `VECTOR` &v)
- `iterator get` () const
- `const_iterator get_const` () const
- `proxy ref` ()
- `proxy ref` (R\_xlen\_t i)
- `proxy ref` () const
- `proxy ref` (R\_xlen\_t i) const

## Private Attributes

- `iterator start`

### 6.627.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
class Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >
```

Definition at line 29 of file traits.h.

### 6.627.2 Member Typedef Documentation

#### 6.627.2.1 const\_iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r_vector_const_iterator<RTYPE>::type Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy
>::const_iterator
```

Definition at line 33 of file traits.h.

### 6.627.2.2 const\_proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r\_vector\_const\_proxy<RTYPE>::type Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >↔
::const_proxy
```

Definition at line 35 of file traits.h.

### 6.627.2.3 iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r\_vector\_iterator<RTYPE>::type Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >↔
::iterator
```

Definition at line 32 of file traits.h.

### 6.627.2.4 proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r\_vector\_proxy<RTYPE>::type Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::proxy
```

Definition at line 34 of file traits.h.

### 6.627.2.5 storage\_type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef storage\_type<RTYPE>::type Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::storage_type
```

Definition at line 36 of file traits.h.

### 6.627.2.6 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::VECTOR
```

Definition at line 31 of file traits.h.

## 6.627.3 Constructor & Destructor Documentation

### 6.627.3.1 r\_vector\_cache()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::r_vector_cache ( ) [inline]
```

Definition at line 38 of file traits.h.

## 6.627.4 Member Function Documentation

### 6.627.4.1 get()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::get ( ) const [inline]
```

Definition at line 42 of file traits.h.

References Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::start.

### 6.627.4.2 get\_const()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::get_const ( ) const [inline]
```

Definition at line 43 of file traits.h.

References Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::start.

### 6.627.4.3 ref() [1/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
proxy Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::ref ( ) [inline]
```

Definition at line 45 of file traits.h.

References Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::start.

#### 6.627.4.4 ref() [2/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
proxy Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::ref ( ) const [inline]
```

Definition at line 48 of file traits.h.

References Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::start.

#### 6.627.4.5 ref() [3/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
proxy Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::ref (
    R_xlen_t i ) [inline]
```

Definition at line 46 of file traits.h.

References Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::start.

#### 6.627.4.6 ref() [4/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
proxy Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::ref (
    R_xlen_t i ) const [inline]
```

Definition at line 49 of file traits.h.

References Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::start.

#### 6.627.4.7 update()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::update (
    const VECTOR & v ) [inline]
```

Definition at line 39 of file traits.h.

References Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::start.

### 6.627.5 Member Data Documentation



### 6.627.5.1 start

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::traits::r_vector_cache< RTYPE, StoragePolicy >::start [private]
```

Definition at line 52 of file traits.h.

Referenced by Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::get(), Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::get\_const(), Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::ref(), and Rcpp::traits::r\_vector\_cache< RTYPE, StoragePolicy >::update().

The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00\_forward\_proxy.h
- inst/include/Rcpp/vector/traits.h

## 6.628 Rcpp::traits::r\_vector\_cache\_type< RTYPE, StoragePolicy > Struct Template Reference

```
#include <traits.h>
```

### Public Types

- typedef [r\\_vector\\_cache](#)< RTYPE, StoragePolicy > [type](#)

### 6.628.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
struct Rcpp::traits::r_vector_cache_type< RTYPE, StoragePolicy >
```

Definition at line 84 of file traits.h.

### 6.628.2 Member Typedef Documentation

#### 6.628.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef r\_vector\_cache<RTYPE, StoragePolicy> Rcpp::traits::r\_vector\_cache\_type< RTYPE, StoragePolicy >::type
```

Definition at line 85 of file traits.h.

The documentation for this struct was generated from the following files:

- inst/include/Rcpp/vector/00\_forward\_proxy.h
- inst/include/Rcpp/vector/traits.h

## 6.629 Rcpp::traits::r\_vector\_cache\_type< EXPRXP, StoragePolicy > Struct Template Reference

```
#include <traits.h>
```

### Public Types

- typedef [proxy\\_cache](#)< EXPRXP, StoragePolicy > [type](#)

### 6.629.1 Detailed Description

```
template<template< class > class StoragePolicy>  
struct Rcpp::traits::r_vector_cache_type< EXPRXP, StoragePolicy >
```

Definition at line 94 of file traits.h.

### 6.629.2 Member Typedef Documentation

#### 6.629.2.1 type

```
template<template< class > class StoragePolicy>  
typedef proxy\_cache<EXPRXP, StoragePolicy> Rcpp::traits::r\_vector\_cache\_type< EXPRXP, Storage↔  
Policy >::type
```

Definition at line 95 of file traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/traits.h](#)

## 6.630 Rcpp::traits::r\_vector\_cache\_type< STRSXP, StoragePolicy > Struct Template Reference

```
#include <traits.h>
```

### Public Types

- typedef [proxy\\_cache](#)< STRSXP, StoragePolicy > [type](#)

### 6.630.1 Detailed Description

```
template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_cache_type< STRSXP, StoragePolicy >
```

Definition at line 98 of file traits.h.

### 6.630.2 Member Typedef Documentation

#### 6.630.2.1 type

```
template<template< class > class StoragePolicy>
typedef proxy_cache<STRSXP, StoragePolicy> Rcpp::traits::r_vector_cache_type< STRSXP, Storage←
Policy >::type
```

Definition at line 99 of file traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/traits.h](#)

## 6.631 Rcpp::traits::r\_vector\_cache\_type< VECSXP, StoragePolicy > Struct Template Reference

```
#include <traits.h>
```

### Public Types

- typedef [proxy\\_cache](#)< VECSXP, StoragePolicy > [type](#)

#### 6.631.1 Detailed Description

```
template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_cache_type< VECSXP, StoragePolicy >
```

Definition at line 90 of file traits.h.

## 6.631.2 Member Typedef Documentation

### 6.631.2.1 type

```
template<template< class > class StoragePolicy>
typedef proxy_cache<VECSXP, StoragePolicy> Rcpp::traits::r_vector_cache_type< VECSXP, StoragePolicy >::type
```

Definition at line 91 of file traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/traits.h](#)

## 6.632 Rcpp::traits::r\_vector\_const\_iterator< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef const [storage\\_type](#)< RTYPE >::type \* type

### 6.632.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_iterator< RTYPE, StoragePolicy >
```

Definition at line 254 of file proxy.h.

## 6.632.2 Member Typedef Documentation

### 6.632.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const storage_type<RTYPE>::type* Rcpp::traits::r_vector_const_iterator< RTYPE, StoragePolicy >::type
```

Definition at line 255 of file proxy.h.

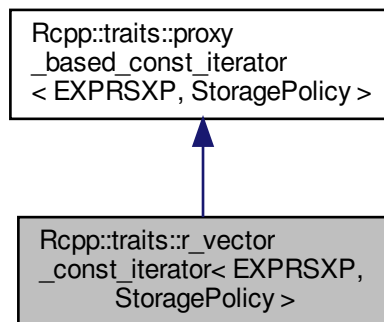
The documentation for this struct was generated from the following files:

- inst/include/Rcpp/vector/00\_forward\_proxy.h
- inst/include/Rcpp/vector/proxy.h

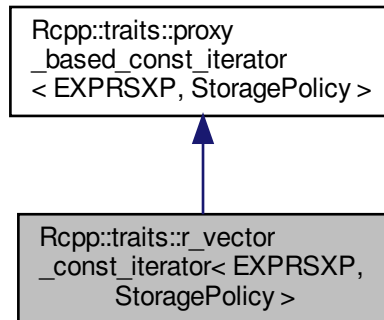
## 6.633 Rcpp::traits::r\_vector\_const\_iterator< EXPRXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Inheritance diagram for Rcpp::traits::r\_vector\_const\_iterator< EXPRXP, StoragePolicy >:



Collaboration diagram for `Rcpp::traits::r_vector_const_iterator< EXPRXP, StoragePolicy >`:



## Additional Inherited Members

### 6.633.1 Detailed Description

```

template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_iterator< EXPRXP, StoragePolicy >

```

Definition at line 271 of file proxy.h.

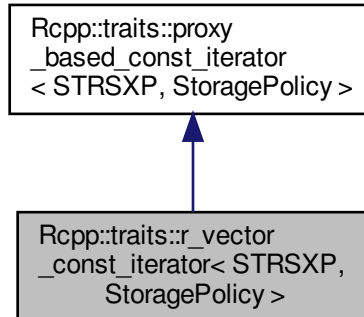
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

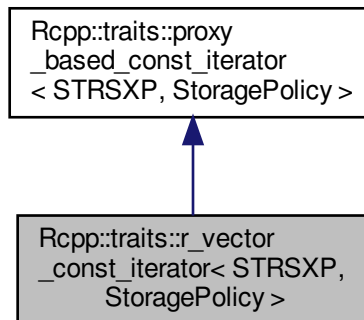
## 6.634 `Rcpp::traits::r_vector_const_iterator< STRXP, StoragePolicy >` Struct Template Reference

```
#include <proxy.h>
```

Inheritance diagram for Rcpp::traits::r\_vector\_const\_iterator< STRSXP, StoragePolicy >:



Collaboration diagram for Rcpp::traits::r\_vector\_const\_iterator< STRSXP, StoragePolicy >:



## Additional Inherited Members

### 6.634.1 Detailed Description

```
template<template< class > class StoragePolicy>  
struct Rcpp::traits::r_vector_const_iterator< STRSXP, StoragePolicy >
```

Definition at line 272 of file proxy.h.

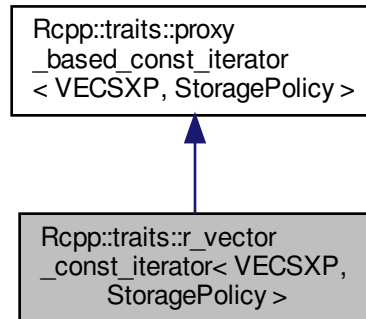
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/[proxy.h](#)

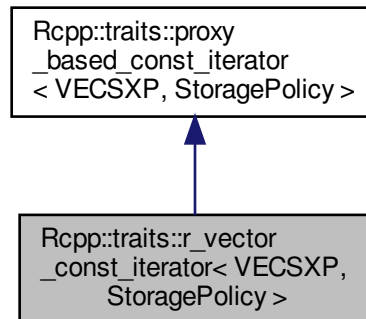
## 6.635 Rcpp::traits::r\_vector\_const\_iterator< VECSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Inheritance diagram for Rcpp::traits::r\_vector\_const\_iterator< VECSXP, StoragePolicy >:



Collaboration diagram for Rcpp::traits::r\_vector\_const\_iterator< VECSXP, StoragePolicy >:



### Additional Inherited Members

#### 6.635.1 Detailed Description



```
template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_iterator< VECSXP, StoragePolicy >
```

Definition at line 270 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

## 6.636 Rcpp::traits::r\_vector\_const\_proxy< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef const [storage\\_type](#)< RTYPE >::type & type

### 6.636.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_proxy< RTYPE, StoragePolicy >
```

Definition at line 233 of file proxy.h.

### 6.636.2 Member Typedef Documentation

#### 6.636.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const storage\_type<RTYPE>::type& Rcpp::traits::r\_vector\_const\_proxy< RTYPE, StoragePolicy
>::type
```

Definition at line 234 of file proxy.h.

The documentation for this struct was generated from the following files:

- inst/include/Rcpp/vector/00\_forward\_proxy.h
- inst/include/Rcpp/vector/proxy.h

## 6.637 Rcpp::traits::r\_vector\_const\_proxy< EXPRXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef [::Rcpp::internal::const\\_generic\\_proxy< EXPRXP, StoragePolicy >](#) [type](#)

### 6.637.1 Detailed Description

```
template<template< class > class StoragePolicy>  
struct Rcpp::traits::r_vector_const_proxy< EXPRXP, StoragePolicy >
```

Definition at line 245 of file proxy.h.

### 6.637.2 Member Typedef Documentation

#### 6.637.2.1 type

```
template<template< class > class StoragePolicy>  
typedef ::Rcpp::internal::const\_generic\_proxy<EXPRXP, StoragePolicy> Rcpp::traits::r\_vector\_const\_proxy<  
EXPRXP, StoragePolicy >::type
```

Definition at line 246 of file proxy.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

## 6.638 Rcpp::traits::r\_vector\_const\_proxy< STRXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef [::Rcpp::internal::const\\_string\\_proxy< STRXP, StoragePolicy >](#) [type](#)

### 6.638.1 Detailed Description

```
template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_proxy< STRSXP, StoragePolicy >
```

Definition at line 237 of file proxy.h.

### 6.638.2 Member Typedef Documentation

#### 6.638.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::const_string_proxy<STRSXP, StoragePolicy> Rcpp::traits::r_vector_const_proxy<
STRSXP, StoragePolicy >::type
```

Definition at line 238 of file proxy.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

## 6.639 Rcpp::traits::r\_vector\_const\_proxy< VECSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef ::Rcpp::internal::const\_generic\_proxy< VECSXP, StoragePolicy > type

### 6.639.1 Detailed Description

```
template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_const_proxy< VECSXP, StoragePolicy >
```

Definition at line 241 of file proxy.h.

## 6.639.2 Member Typedef Documentation

### 6.639.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::const_generic_proxy<VECSXP, StoragePolicy> Rcpp::traits::r_vector_const_proxy<
VECSXP, StoragePolicy >::type
```

Definition at line 242 of file proxy.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

## 6.640 Rcpp::traits::r\_vector\_element\_converter< RTYPE > Struct Template Reference

```
#include <converter.h>
```

### Public Types

- typedef ::Rcpp::internal::element\_converter< RTYPE > type

### 6.640.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::traits::r_vector_element_converter< RTYPE >
```

Definition at line 94 of file converter.h.

### 6.640.2 Member Typedef Documentation

### 6.640.2.1 type

```
template<int RTYPE>
typedef ::Rcpp::internal::element_converter<RTYPE> Rcpp::traits::r_vector_element_converter<
RTYPE >::type
```

Definition at line 95 of file converter.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/[converter.h](#)

## 6.641 Rcpp::traits::r\_vector\_element\_converter< EXPRXP > Struct Reference

```
#include <converter.h>
```

### Public Types

- typedef ::Rcpp::internal::generic\_element\_converter< EXPRXP > type

### 6.641.1 Detailed Description

Definition at line 103 of file converter.h.

### 6.641.2 Member Typedef Documentation

#### 6.641.2.1 type

```
typedef ::Rcpp::internal::generic_element_converter<EXPRXP> Rcpp::traits::r_vector_element_converter<
EXPRXP >::type
```

Definition at line 104 of file converter.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/[converter.h](#)

## 6.642 Rcpp::traits::r\_vector\_element\_converter< STRSXP > Struct Reference

```
#include <converter.h>
```

### Public Types

- typedef [::Rcpp::internal::string\\_element\\_converter< STRSXP >](#) type

### 6.642.1 Detailed Description

Definition at line 97 of file converter.h.

### 6.642.2 Member Typedef Documentation

#### 6.642.2.1 type

```
typedef ::Rcpp::internal::string\_element\_converter<STRSXP> Rcpp::traits::r\_vector\_element\_converter<STRSXP >::type
```

Definition at line 98 of file converter.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/converter.h](#)

## 6.643 Rcpp::traits::r\_vector\_element\_converter< VECSXP > Struct Reference

```
#include <converter.h>
```

### Public Types

- typedef [::Rcpp::internal::generic\\_element\\_converter< VECSXP >](#) type

### 6.643.1 Detailed Description

Definition at line 100 of file converter.h.

## 6.643.2 Member Typedef Documentation

### 6.643.2.1 type

```
typedef ::Rcpp::internal::generic_element_converter<VECSXP> Rcpp::traits::r_vector_element_converter<VECSXP >::type
```

Definition at line 101 of file converter.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/[converter.h](#)

## 6.644 Rcpp::traits::r\_vector\_iterator< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef [storage\\_type](#)< RTYPE >::type \* type

### 6.644.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>  
struct Rcpp::traits::r_vector_iterator< RTYPE, StoragePolicy >
```

Definition at line 250 of file proxy.h.

## 6.644.2 Member Typedef Documentation

### 6.644.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>  
typedef storage\_type<RTYPE>::type* Rcpp::traits::r_vector_iterator< RTYPE, StoragePolicy >::type
```

Definition at line 251 of file proxy.h.

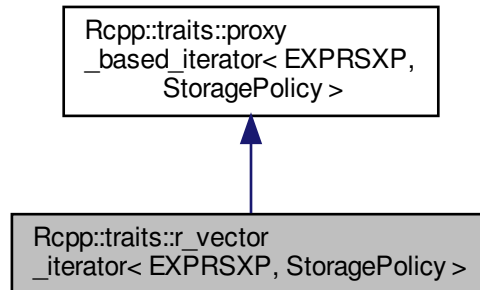
The documentation for this struct was generated from the following files:

- inst/include/Rcpp/vector/[00\\_forward\\_proxy.h](#)
- inst/include/Rcpp/vector/[proxy.h](#)

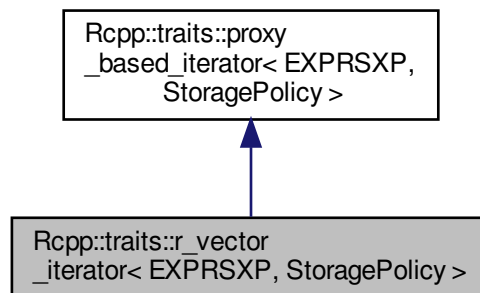
## 6.645 Rcpp::traits::r\_vector\_iterator< EXPRXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Inheritance diagram for Rcpp::traits::r\_vector\_iterator< EXPRXP, StoragePolicy >:



Collaboration diagram for Rcpp::traits::r\_vector\_iterator< EXPRXP, StoragePolicy >:



### Additional Inherited Members

#### 6.645.1 Detailed Description



```
template<template< class > class StoragePolicy>  
struct Rcpp::traits::r_vector_iterator< EXPRSXP, StoragePolicy >
```

Definition at line 263 of file proxy.h.

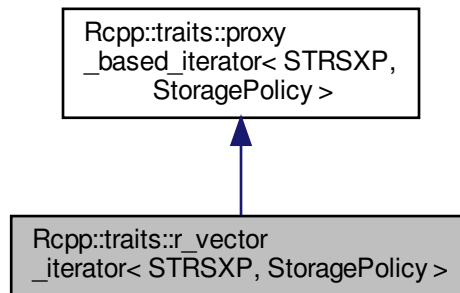
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

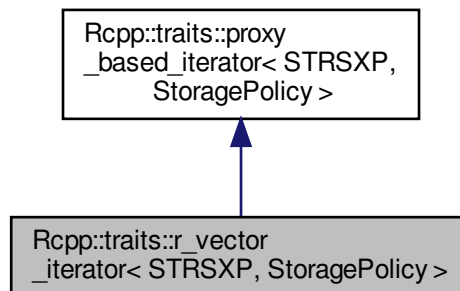
## 6.646 Rcpp::traits::r\_vector\_iterator< STRSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Inheritance diagram for Rcpp::traits::r\_vector\_iterator< STRSXP, StoragePolicy >:



Collaboration diagram for Rcpp::traits::r\_vector\_iterator< STRSXP, StoragePolicy >:



## Additional Inherited Members

### 6.646.1 Detailed Description

```
template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_iterator< STRSXP, StoragePolicy >
```

Definition at line 264 of file proxy.h.

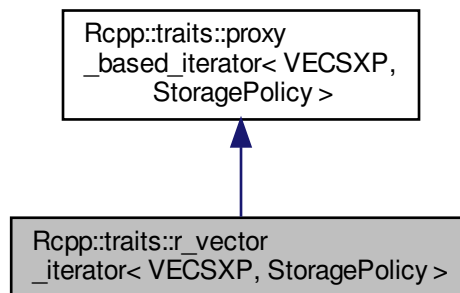
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

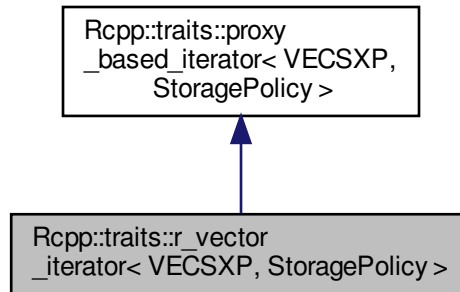
### 6.647 Rcpp::traits::r\_vector\_iterator< VECSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

Inheritance diagram for Rcpp::traits::r\_vector\_iterator< VECSXP, StoragePolicy >:



Collaboration diagram for Rcpp::traits::r\_vector\_iterator< VECSXP, StoragePolicy >:



## Additional Inherited Members

### 6.647.1 Detailed Description

```
template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_iterator< VECSXP, StoragePolicy >
```

Definition at line 262 of file proxy.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

## 6.648 Rcpp::traits::r\_vector\_name\_proxy< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef [::Rcpp::internal::simple\\_name\\_proxy](#)< RTYPE, StoragePolicy > [type](#)

### 6.648.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_name_proxy< RTYPE, StoragePolicy >
```

Definition at line 199 of file proxy.h.

### 6.648.2 Member Typedef Documentation

#### 6.648.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::internal::simple_name_proxy<RTYPE, StoragePolicy> Rcpp::traits::r_vector_name_proxy<
RTYPE, StoragePolicy >::type
```

Definition at line 200 of file proxy.h.

The documentation for this struct was generated from the following files:

- [inst/include/Rcpp/vector/00\\_forward\\_proxy.h](#)
- [inst/include/Rcpp/vector/proxy.h](#)

## 6.649 Rcpp::traits::r\_vector\_name\_proxy< EXPRXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef [::Rcpp::internal::generic\\_name\\_proxy< EXPRXP, StoragePolicy >](#) [type](#)

### 6.649.1 Detailed Description

```
template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_name_proxy< EXPRXP, StoragePolicy >
```

Definition at line 211 of file proxy.h.

## 6.649.2 Member Typedef Documentation

### 6.649.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::generic_name_proxy<EXPRSXP, StoragePolicy> Rcpp::traits::r_vector_name_proxy<
EXPRSXP, StoragePolicy >::type
```

Definition at line 212 of file proxy.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/proxy.h

## 6.650 Rcpp::traits::r\_vector\_name\_proxy< STRSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef ::Rcpp::internal::string\_name\_proxy< STRSXP, StoragePolicy > type

### 6.650.1 Detailed Description

```
template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_name_proxy< STRSXP, StoragePolicy >
```

Definition at line 203 of file proxy.h.

### 6.650.2 Member Typedef Documentation

### 6.650.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::string_name_proxy<STRSXP, StoragePolicy> Rcpp::traits::r_vector_name_proxy<
STRSXP, StoragePolicy >::type
```

Definition at line 204 of file proxy.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

## 6.651 Rcpp::traits::r\_vector\_name\_proxy< VECSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef ::Rcpp::internal::generic\_name\_proxy< VECSXP, StoragePolicy > type

### 6.651.1 Detailed Description

```
template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_name_proxy< VECSXP, StoragePolicy >
```

Definition at line 207 of file proxy.h.

### 6.651.2 Member Typedef Documentation

#### 6.651.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::generic_name_proxy<VECSXP, StoragePolicy> Rcpp::traits::r_vector_name_proxy<
VECSXP, StoragePolicy >::type
```

Definition at line 208 of file proxy.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

## 6.652 Rcpp::traits::r\_vector\_proxy< RTYPE, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef [storage\\_type](#)< RTYPE >::type & [type](#)

### 6.652.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>  
struct Rcpp::traits::r_vector_proxy< RTYPE, StoragePolicy >
```

Definition at line 216 of file proxy.h.

### 6.652.2 Member Typedef Documentation

#### 6.652.2.1 type

```
template<int RTYPE, template< class > class StoragePolicy>  
typedef storage\_type<RTYPE>::type& Rcpp::traits::r\_vector\_proxy< RTYPE, StoragePolicy >::type
```

Definition at line 217 of file proxy.h.

The documentation for this struct was generated from the following files:

- inst/include/Rcpp/vector/00\_forward\_proxy.h
- inst/include/Rcpp/vector/proxy.h

## 6.653 Rcpp::traits::r\_vector\_proxy< EXPRXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef [::Rcpp::internal::generic\\_proxy](#)< EXPRXP, StoragePolicy > [type](#)

### 6.653.1 Detailed Description

```
template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_proxy< EXPRSXP, StoragePolicy >
```

Definition at line 224 of file proxy.h.

### 6.653.2 Member Typedef Documentation

#### 6.653.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::generic_proxy<EXPRSXP, StoragePolicy> Rcpp::traits::r_vector_proxy<
EXPRSXP, StoragePolicy >::type
```

Definition at line 225 of file proxy.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

## 6.654 Rcpp::traits::r\_vector\_proxy< STRSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef ::Rcpp::internal::string\_proxy< STRSXP, StoragePolicy > type

#### 6.654.1 Detailed Description

```
template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_proxy< STRSXP, StoragePolicy >
```

Definition at line 220 of file proxy.h.



## 6.654.2 Member Typedef Documentation

### 6.654.2.1 type

```
template<template< class > class StoragePolicy>
typedef ::Rcpp::internal::string_proxy<STRSXP, StoragePolicy> Rcpp::traits::r_vector_proxy< STRSXP,
StoragePolicy >::type
```

Definition at line 221 of file proxy.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

## 6.655 Rcpp::traits::r\_vector\_proxy< VECSXP, StoragePolicy > Struct Template Reference

```
#include <proxy.h>
```

### Public Types

- typedef ::Rcpp::internal::generic\_proxy< VECSXP, StoragePolicy > [type](#)

### 6.655.1 Detailed Description

```
template<template< class > class StoragePolicy>
struct Rcpp::traits::r_vector_proxy< VECSXP, StoragePolicy >
```

Definition at line 228 of file proxy.h.

## 6.655.2 Member Typedef Documentation

### 6.655.2.1 type

```
template<template< class > class StoragePolicy>  
typedef ::Rcpp::internal::generic_proxy<VECSXP, StoragePolicy> Rcpp::traits::r_vector_proxy<  
VECSXP, StoragePolicy >::type
```

Definition at line 229 of file proxy.h.

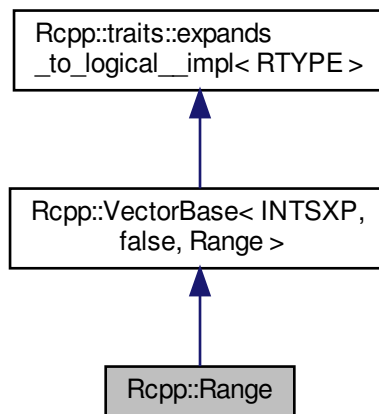
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/vector/proxy.h](#)

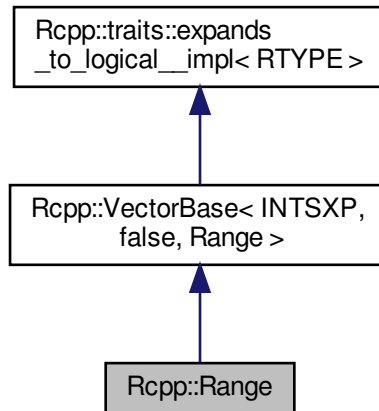
## 6.656 Rcpp::Range Class Reference

```
#include <Range.h>
```

Inheritance diagram for Rcpp::Range:



Collaboration diagram for Rcpp::Range:



## Public Member Functions

- [Range](#) (R\_xlen\_t start\_, R\_xlen\_t end\_)
- R\_xlen\_t [size](#) () const
- R\_xlen\_t [operator\[\]](#) (R\_xlen\_t i) const
- [Range & operator++](#) ()
- [Range operator++](#) (int)
- [Range & operator--](#) ()
- [Range operator--](#) (int)
- [Range & operator+=](#) (int n)
- [Range & operator-=](#) (int n)
- [Range operator+](#) (int n)
- [Range operator-](#) (int n)
- R\_xlen\_t [get\\_start](#) () const
- R\_xlen\_t [get\\_end](#) () const

## Private Attributes

- R\_xlen\_t [start](#)
- R\_xlen\_t [end\\_](#)

## Additional Inherited Members

### 6.656.1 Detailed Description

Definition at line 27 of file Range.h.

## 6.656.2 Constructor & Destructor Documentation

### 6.656.2.1 Range()

```
Rcpp::Range::Range (
    R_xlen_t start_,
    R_xlen_t end_ ) [inline]
```

Definition at line 29 of file Range.h.

Referenced by operator+(), and operator-().

## 6.656.3 Member Function Documentation

### 6.656.3.1 get\_end()

```
R_xlen_t Rcpp::Range::get_end ( ) const [inline]
```

Definition at line 83 of file Range.h.

References end\_.

### 6.656.3.2 get\_start()

```
R_xlen_t Rcpp::Range::get_start ( ) const [inline]
```

Definition at line 81 of file Range.h.

References start.

### 6.656.3.3 operator+()

```
Range Rcpp::Range::operator+ (
    int n ) [inline]
```

Definition at line 73 of file Range.h.

References `end_`, `Range()`, and `start`.

Here is the call graph for this function:



### 6.656.3.4 operator++() [1/2]

```
Range& Rcpp::Range::operator++ ( ) [inline]
```

Definition at line 43 of file Range.h.

References `end_`, and `start`.

### 6.656.3.5 operator++() [2/2]

```
Range Rcpp::Range::operator++ (
    int ) [inline]
```

Definition at line 47 of file Range.h.

### 6.656.3.6 operator+=()

```
Range& Rcpp::Range::operator+= (
    int n ) [inline]
```

Definition at line 63 of file Range.h.

References `end_`, and `start`.

### 6.656.3.7 operator-()

```
Range Rcpp::Range::operator- (
    int n ) [inline]
```

Definition at line 77 of file Range.h.

References `end_`, `Range()`, and `start`.

Here is the call graph for this function:



### 6.656.3.8 operator--() [1/2]

```
Range& Rcpp::Range::operator-- ( ) [inline]
```

Definition at line 53 of file Range.h.

References `end_`, and `start`.

### 6.656.3.9 operator--() [2/2]

```
Range Rcpp::Range::operator-- (
    int ) [inline]
```

Definition at line 57 of file Range.h.

### 6.656.3.10 operator==()

```
Range& Rcpp::Range::operator== (
    int n ) [inline]
```

Definition at line 68 of file Range.h.

References `end_`, and `start`.

### 6.656.3.11 operator[]()

```
R_xlen_t Rcpp::Range::operator[] (
    R_xlen_t i) const [inline]
```

Definition at line 39 of file Range.h.

References start.

### 6.656.3.12 size()

```
R_xlen_t Rcpp::Range::size ( ) const [inline]
```

Definition at line 35 of file Range.h.

References end\_, and start.

## 6.656.4 Member Data Documentation

### 6.656.4.1 end\_

```
R_xlen_t Rcpp::Range::end_ [private]
```

Definition at line 87 of file Range.h.

Referenced by get\_end(), operator+(), operator++(), operator+=(), operator-(), operator--(), operator-=(), and size().

### 6.656.4.2 start

```
R_xlen_t Rcpp::Range::start [private]
```

Definition at line 86 of file Range.h.

Referenced by get\_start(), operator+(), operator++(), operator+=(), operator-(), operator--(), operator-=(), operator[](), and size().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/[Range.h](#)

## 6.657 Rcpp::sugar::Range< RTYPE, NA, T > Class Template Reference

```
#include <range.h>
```

### Public Types

- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)

### Public Member Functions

- [Range](#) (const T &obj\_)
- [operator Vector](#)< RTYPE > ()

### Private Attributes

- const T & [obj](#)
- [STORAGE](#) [min\\_](#)
- [STORAGE](#) [max\\_](#)
- [STORAGE](#) [current](#)

### 6.657.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::Range< RTYPE, NA, T >
```

Definition at line 29 of file range.h.

### 6.657.2 Member Typedef Documentation

#### 6.657.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Range< RTYPE, NA, T >::STORAGE
```

Definition at line 31 of file range.h.

### 6.657.3 Constructor & Destructor Documentation



### 6.657.3.1 Range()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Range< RTYPE, NA, T >::Range (
    const T & obj_ ) [inline]
```

Definition at line 33 of file range.h.

## 6.657.4 Member Function Documentation

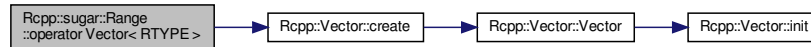
### 6.657.4.1 operator Vector< RTYPE >()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Range< RTYPE, NA, T >::operator Vector< RTYPE > ( ) [inline]
```

Definition at line 35 of file range.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::create(), Rcpp::sugar::Range< RTYPE, NA, T >::current, Rcpp::sugar::Range< RTYPE, NA, T >::max\_, Rcpp::sugar::Range< RTYPE, NA, T >::min\_, and Rcpp::sugar::Range< RTYPE, NA, T >::obj.

Here is the call graph for this function:



## 6.657.5 Member Data Documentation

### 6.657.5.1 current

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Range< RTYPE, NA, T >::current [private]
```

Definition at line 53 of file range.h.

Referenced by Rcpp::sugar::Range< RTYPE, NA, T >::operator Vector< RTYPE >(), and Rcpp::sugar::Range< RTYPE, false, T >::operator Vector< RTYPE >().

### 6.657.5.2 max\_

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Range< RTYPE, NA, T >::max_ [private]
```

Definition at line 53 of file range.h.

Referenced by `Rcpp::sugar::Range< RTYPE, NA, T >::operator Vector< RTYPE >()`, and `Rcpp::sugar::Range< RTYPE, false, T >::operator Vector< RTYPE >()`.

### 6.657.5.3 min\_

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Range< RTYPE, NA, T >::min_ [private]
```

Definition at line 53 of file range.h.

Referenced by `Rcpp::sugar::Range< RTYPE, NA, T >::operator Vector< RTYPE >()`, and `Rcpp::sugar::Range< RTYPE, false, T >::operator Vector< RTYPE >()`.

### 6.657.5.4 obj

```
template<int RTYPE, bool NA, typename T >
const T& Rcpp::sugar::Range< RTYPE, NA, T >::obj [private]
```

Definition at line 52 of file range.h.

Referenced by `Rcpp::sugar::Range< RTYPE, NA, T >::operator Vector< RTYPE >()`, and `Rcpp::sugar::Range< RTYPE, false, T >::operator Vector< RTYPE >()`.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/range.h](#)

## 6.658 Rcpp::sugar::Range< RTYPE, false, T > Class Template Reference

```
#include <range.h>
```

### Public Types

- typedef `Rcpp::traits::storage_type< RTYPE >::type STORAGE`

## Public Member Functions

- [Range](#) (const T &obj\_)
- [operator Vector< RTYPE > \(\)](#)

## Private Attributes

- const T & [obj](#)
- [STORAGE](#) [min\\_](#)
- [STORAGE](#) [max\\_](#)
- [STORAGE](#) [current](#)

### 6.658.1 Detailed Description

```
template<int RTYPE, typename T>
class Rcpp::sugar::Range< RTYPE, false, T >
```

Definition at line 58 of file range.h.

### 6.658.2 Member Typedef Documentation

#### 6.658.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Range< RTYPE, false, T >::STORAGE
```

Definition at line 60 of file range.h.

### 6.658.3 Constructor & Destructor Documentation

#### 6.658.3.1 Range()

```
template<int RTYPE, typename T >
Rcpp::sugar::Range< RTYPE, false, T >::Range (
    const T & obj\_ ) [inline]
```

Definition at line 62 of file range.h.

## 6.658.4 Member Function Documentation

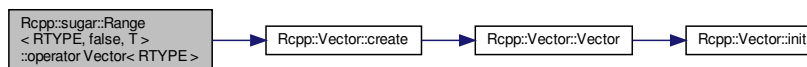
### 6.658.4.1 operator Vector< RTYPE >()

```
template<int RTYPE, typename T >
Rcpp::sugar::Range< RTYPE, false, T >::operator Vector< RTYPE > ( ) [inline]
```

Definition at line 64 of file range.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::create(), Rcpp::sugar::Range< RTYPE, NA, T >::current, Rcpp::sugar::Range< RTYPE, NA, T >::max\_, Rcpp::sugar::Range< RTYPE, NA, T >::min\_, and Rcpp::sugar::Range< RTYPE, NA, T >::obj.

Here is the call graph for this function:



## 6.658.5 Member Data Documentation

### 6.658.5.1 current

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Range< RTYPE, false, T >::current [private]
```

Definition at line 79 of file range.h.

### 6.658.5.2 max\_

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Range< RTYPE, false, T >::max_ [private]
```

Definition at line 79 of file range.h.

**6.658.5.3 min\_**

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Range< RTYPE, false, T >::min_ [private]
```

Definition at line 79 of file range.h.

**6.658.5.4 obj**

```
template<int RTYPE, typename T >
const T& Rcpp::sugar::Range< RTYPE, false, T >::obj [private]
```

Definition at line 78 of file range.h.

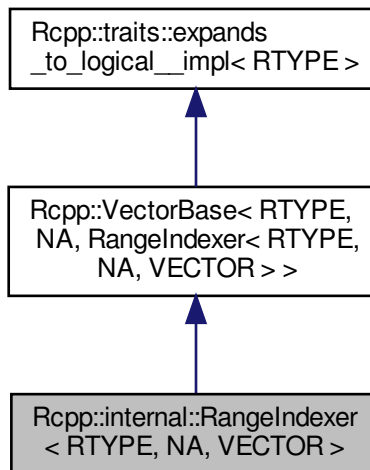
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/range.h

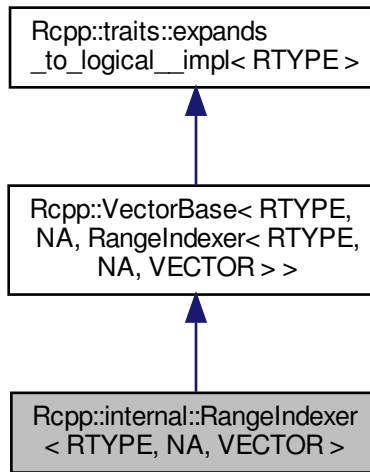
**6.659 Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR > Class Template Reference**

```
#include <RangeIndexer.h>
```

Inheritance diagram for Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >:



Collaboration diagram for `Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >`:



## Public Types

- typedef `VECTOR::Proxy` [Proxy](#)
- typedef `VECTOR::iterator` [iterator](#)

## Public Member Functions

- [RangeIndexer](#) (`VECTOR &vec_`, const [Rcpp::Range](#) &range\_)
- template<bool NA\_, typename T >  
[RangeIndexer](#) & [operator=](#) (const [Rcpp::VectorBase](#)< RTYPE, NA\_, T > &x)
- template<bool NA\_, typename T >  
[RangeIndexer](#) & [operator+=](#) (const [Rcpp::VectorBase](#)< RTYPE, NA\_, T > &x)
- template<bool NA\_, typename T >  
[RangeIndexer](#) & [operator\\*=\[operator\\\*=\]\(#\)](#) (const [Rcpp::VectorBase](#)< RTYPE, NA\_, T > &x)
- template<bool NA\_, typename T >  
[RangeIndexer](#) & [operator-=](#) (const [Rcpp::VectorBase](#)< RTYPE, NA\_, T > &x)
- template<bool NA\_, typename T >  
[RangeIndexer](#) & [operator/=](#) (const [Rcpp::VectorBase](#)< RTYPE, NA\_, T > &x)
- [Proxy operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- [iterator](#) [start](#)
- R\_xlen\_t [size\\_](#)

## 6.659.1 Detailed Description

```
template<int RTYPE, bool NA, typename VECTOR>
class Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >
```

Definition at line 53 of file RangeIndexer.h.

## 6.659.2 Member Typedef Documentation

### 6.659.2.1 iterator

```
template<int RTYPE, bool NA, typename VECTOR >
typedef VECTOR::iterator Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::iterator
```

Definition at line 56 of file RangeIndexer.h.

### 6.659.2.2 Proxy

```
template<int RTYPE, bool NA, typename VECTOR >
typedef VECTOR::Proxy Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::Proxy
```

Definition at line 55 of file RangeIndexer.h.

## 6.659.3 Constructor & Destructor Documentation

### 6.659.3.1 RangeIndexer()

```
template<int RTYPE, bool NA, typename VECTOR >
Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::RangeIndexer (
    VECTOR & vec_,
    const Rcpp::Range & range_ ) [inline]
```

Definition at line 58 of file RangeIndexer.h.

## 6.659.4 Member Function Documentation

#### 6.659.4.1 operator\*=( )

```
template<int RTYPE, bool NA, typename VECTOR >
template<bool NA_, typename T >
RangeIndexer& Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator*= (
    const Rcpp::VectorBase< RTYPE, NA_, T > & x ) [inline]
```

Definition at line 72 of file RangeIndexer.h.

References UNROLL\_LOOP.

#### 6.659.4.2 operator+=( )

```
template<int RTYPE, bool NA, typename VECTOR >
template<bool NA_, typename T >
RangeIndexer& Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator+= (
    const Rcpp::VectorBase< RTYPE, NA_, T > & x ) [inline]
```

Definition at line 67 of file RangeIndexer.h.

References UNROLL\_LOOP.

#### 6.659.4.3 operator--( )

```
template<int RTYPE, bool NA, typename VECTOR >
template<bool NA_, typename T >
RangeIndexer& Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator--= (
    const Rcpp::VectorBase< RTYPE, NA_, T > & x ) [inline]
```

Definition at line 77 of file RangeIndexer.h.

References UNROLL\_LOOP.

#### 6.659.4.4 operator/=( )

```
template<int RTYPE, bool NA, typename VECTOR >
template<bool NA_, typename T >
RangeIndexer& Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator/= (
    const Rcpp::VectorBase< RTYPE, NA_, T > & x ) [inline]
```

Definition at line 82 of file RangeIndexer.h.

References UNROLL\_LOOP.



#### 6.659.4.5 operator=()

```
template<int RTYPE, bool NA, typename VECTOR >
template<bool NA_, typename T >
RangeIndexer& Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator= (
    const Rcpp::VectorBase< RTYPE, NA_, T > & x ) [inline]
```

Definition at line 62 of file RangeIndexer.h.

References UNROLL\_LOOP.

#### 6.659.4.6 operator[]()

```
template<int RTYPE, bool NA, typename VECTOR >
Proxy Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 86 of file RangeIndexer.h.

References Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::start.

#### 6.659.4.7 size()

```
template<int RTYPE, bool NA, typename VECTOR >
R_xlen_t Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::size ( ) const [inline]
```

Definition at line 90 of file RangeIndexer.h.

References Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::size\_.

### 6.659.5 Member Data Documentation

#### 6.659.5.1 size\_

```
template<int RTYPE, bool NA, typename VECTOR >
R_xlen_t Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::size_ [private]
```

Definition at line 96 of file RangeIndexer.h.

Referenced by Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::size().

### 6.659.5.2 start

```
template<int RTYPE, bool NA, typename VECTOR >
iterator Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::start [private]
```

Definition at line 95 of file RangeIndexer.h.

Referenced by Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >::operator[]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/vector/[RangeIndexer.h](#)

## 6.660 Rcpp::VectorBase< RTYPE, na, VECTOR >::rcpp\_sugar\_expression Struct Reference

```
#include <VectorBase.h>
```

### 6.660.1 Detailed Description

```
template<int RTYPE, bool na, typename VECTOR>
struct Rcpp::VectorBase< RTYPE, na, VECTOR >::rcpp_sugar_expression
```

Definition at line 31 of file VectorBase.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/vector/[VectorBase.h](#)

## 6.661 Rcpp::ReferenceInputParameter< T > Class Template Reference

```
#include <InputParameter.h>
```

### Public Types

- typedef T & [reference](#)

### Public Member Functions

- [ReferenceInputParameter](#) (SEXP x\_)
- [operator reference](#) ()

## Private Attributes

- [T obj](#)

### 6.661.1 Detailed Description

```
template<typename T>  
class Rcpp::ReferenceInputParameter< T >
```

Definition at line 43 of file InputParameter.h.

### 6.661.2 Member Typedef Documentation

#### 6.661.2.1 reference

```
template<typename T >  
typedef T& Rcpp::ReferenceInputParameter< T >::reference
```

Definition at line 45 of file InputParameter.h.

### 6.661.3 Constructor & Destructor Documentation

#### 6.661.3.1 ReferenceInputParameter()

```
template<typename T >  
Rcpp::ReferenceInputParameter< T >::ReferenceInputParameter (  
    SEXP x_ ) [inline]
```

Definition at line 46 of file InputParameter.h.

### 6.661.4 Member Function Documentation

### 6.661.4.1 operator reference()

```
template<typename T >  
Rcpp::ReferenceInputParameter< T >::operator reference ( ) [inline]
```

Definition at line 48 of file InputParameter.h.

References Rcpp::ReferenceInputParameter< T >::obj.

## 6.661.5 Member Data Documentation

### 6.661.5.1 obj

```
template<typename T >  
T Rcpp::ReferenceInputParameter< T >::obj [private]
```

Definition at line 51 of file InputParameter.h.

Referenced by Rcpp::ReferenceInputParameter< T >::operator reference().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/InputParameter.h](#)

## 6.662 Rcpp::traits::remove\_const< \_Tp > Struct Template Reference

const-volatile modifications [4.7.1].

```
#include <remove_const.h>
```

### Public Types

- typedef [\\_Tp](#) [type](#)

### 6.662.1 Detailed Description

```
template<typename _Tp>  
struct Rcpp::traits::remove_const< _Tp >
```

const-volatile modifications [4.7.1].

Definition at line 30 of file remove\_const.h.

## 6.662.2 Member Typedef Documentation

### 6.662.2.1 type

```
template<typename _Tp >  
typedef _Tp Rcpp::traits::remove_const< _Tp >::type
```

Definition at line 31 of file remove\_const.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/remove\_const.h

## 6.663 Rcpp::traits::remove\_const< \_Tp const > Struct Template Reference

```
#include <remove_const.h>
```

### Public Types

- typedef \_Tp [type](#)

### 6.663.1 Detailed Description

```
template<typename _Tp>  
struct Rcpp::traits::remove_const< _Tp const >
```

Definition at line 34 of file remove\_const.h.

## 6.663.2 Member Typedef Documentation

### 6.663.2.1 type

```
template<typename _Tp >  
typedef _Tp Rcpp::traits::remove_const< _Tp const >::type
```

Definition at line 35 of file remove\_const.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/remove\_const.h

## 6.664 Rcpp::traits::remove\_const\_and\_reference< T > Struct Template Reference

```
#include <remove_const_and_reference.h>
```

### Public Types

- typedef [remove\\_const](#)< typename [remove\\_reference](#)< T >::type >::type [type](#)

### 6.664.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::remove_const_and_reference< T >
```

Definition at line 29 of file [remove\\_const\\_and\\_reference.h](#).

### 6.664.2 Member Typedef Documentation

#### 6.664.2.1 type

```
template<typename T >  
typedef remove\_const< typename remove\_reference<T>::type >::type Rcpp::traits::remove\_const\_and\_reference<  
T >::type
```

Definition at line 30 of file [remove\\_const\\_and\\_reference.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/remove\\_const\\_and\\_reference.h](#)

## 6.665 Rcpp::traits::remove\_reference< \_Tp > Struct Template Reference

reference modifications [4.7.2].

```
#include <remove_reference.h>
```

### Public Types

- typedef [\\_Tp](#) [type](#)

### 6.665.1 Detailed Description

```
template<typename _Tp>
struct Rcpp::traits::remove_reference< _Tp >
```

reference modifications [4.7.2].

Definition at line 30 of file remove\_reference.h.

### 6.665.2 Member Typedef Documentation

#### 6.665.2.1 type

```
template<typename _Tp >
typedef _Tp Rcpp::traits::remove_reference< _Tp >::type
```

Definition at line 31 of file remove\_reference.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[remove\\_reference.h](#)

## 6.666 Rcpp::traits::remove\_reference< \_Tp & > Struct Template Reference

```
#include <remove_reference.h>
```

### Public Types

- typedef \_Tp [type](#)

### 6.666.1 Detailed Description

```
template<typename _Tp>
struct Rcpp::traits::remove_reference< _Tp & >
```

Definition at line 34 of file remove\_reference.h.

### 6.666.2 Member Typedef Documentation

### 6.666.2.1 type

```
template<typename _Tp >
typedef _Tp Rcpp::traits::remove_reference< _Tp & >::type
```

Definition at line 35 of file `remove_reference.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/remove_reference.h`

## 6.667 Rcpp::sugar::RemoveFromSet< SET > Class Template Reference

```
#include <setdiff.h>
```

### Public Member Functions

- [RemoveFromSet](#) (SET &set\_)
- `template<typename T >`  
void [operator\(\)](#) (T value)

### Private Attributes

- SET & [set](#)

### 6.667.1 Detailed Description

```
template<typename SET>
class Rcpp::sugar::RemoveFromSet< SET >
```

Definition at line 29 of file `setdiff.h`.

### 6.667.2 Constructor & Destructor Documentation

#### 6.667.2.1 RemoveFromSet()

```
template<typename SET >
Rcpp::sugar::RemoveFromSet< SET >::RemoveFromSet (
    SET & set_ ) [inline]
```

Definition at line 31 of file `setdiff.h`.



## 6.667.3 Member Function Documentation

### 6.667.3.1 operator>()

```
template<typename SET >
template<typename T >
void Rcpp::sugar::RemoveFromSet< SET >::operator() (
    T value ) [inline]
```

Definition at line 34 of file setdiff.h.

References Rcpp::sugar::RemoveFromSet< SET >::set.

## 6.667.4 Member Data Documentation

### 6.667.4.1 set

```
template<typename SET >
SET& Rcpp::sugar::RemoveFromSet< SET >::set [private]
```

Definition at line 39 of file setdiff.h.

Referenced by Rcpp::sugar::RemoveFromSet< SET >::operator>().

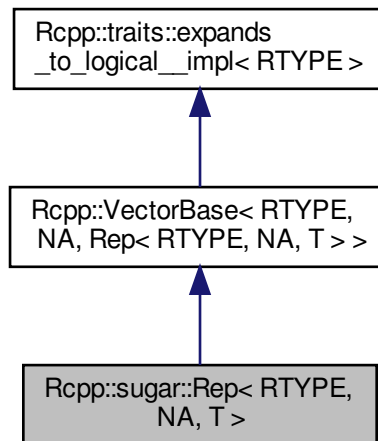
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[setdiff.h](#)

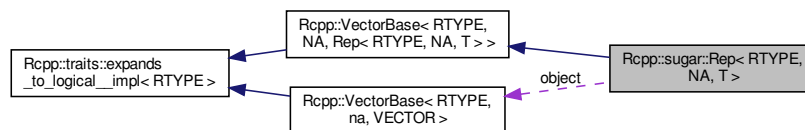
## 6.668 Rcpp::sugar::Rep< RTYPE, NA, T > Class Template Reference

```
#include <rep.h>
```

Inheritance diagram for Rcpp::sugar::Rep< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Rep< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`

### Public Member Functions

- `Rep` (const `VEC_TYPE` &object\_, `R_xlen_t` times\_)
- `STORAGE` operator[] (`R_xlen_t` i) const
- `R_xlen_t` size () const

## Private Attributes

- const [VEC\\_TYPE](#) & *object*
- `R_xlen_t` *times*
- `R_xlen_t` *n*

### 6.668.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Rep< RTYPE, NA, T >
```

Definition at line 29 of file rep.h.

### 6.668.2 Member Typedef Documentation

#### 6.668.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Rep< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file rep.h.

#### 6.668.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Rep< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file rep.h.

### 6.668.3 Constructor & Destructor Documentation

#### 6.668.3.1 Rep()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Rep< RTYPE, NA, T >::Rep (
    const VEC\_TYPE & object_,
    R_xlen_t times_ ) [inline]
```

Definition at line 34 of file rep.h.

## 6.668.4 Member Function Documentation

### 6.668.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >  
STORAGE Rcpp::sugar::Rep< RTYPE, NA, T >::operator[] (   
    R_xlen_t i ) const [inline]
```

Definition at line 37 of file rep.h.

References `Rcpp::sugar::Rep< RTYPE, NA, T >::n`.

### 6.668.4.2 size()

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Rep< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 40 of file rep.h.

References `Rcpp::sugar::Rep< RTYPE, NA, T >::n`, and `Rcpp::sugar::Rep< RTYPE, NA, T >::times`.

## 6.668.5 Member Data Documentation

### 6.668.5.1 n

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Rep< RTYPE, NA, T >::n [private]
```

Definition at line 44 of file rep.h.

Referenced by `Rcpp::sugar::Rep< RTYPE, NA, T >::operator[]()`, and `Rcpp::sugar::Rep< RTYPE, NA, T >::size()`.

### 6.668.5.2 object

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::sugar::Rep< RTYPE, NA, T >::object [private]
```

Definition at line 43 of file rep.h.

## 6.668.5.3 times

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rep< RTYPE, NA, T >::times [private]
```

Definition at line 44 of file rep.h.

Referenced by Rcpp::sugar::Rep< RTYPE, NA, T >::size().

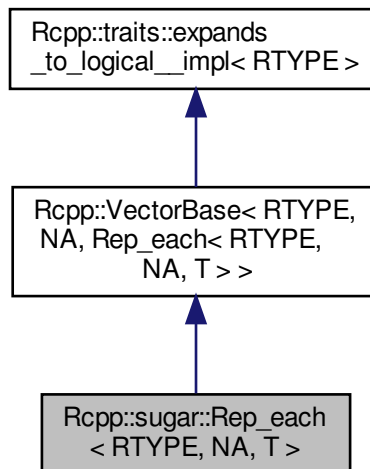
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rep.h

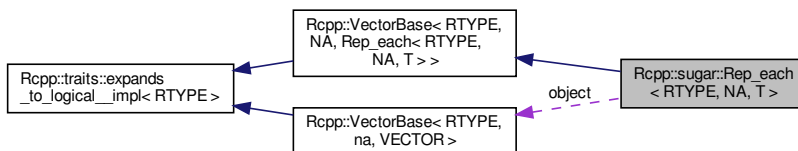
## 6.669 Rcpp::sugar::Rep\_each&lt; RTYPE, NA, T &gt; Class Template Reference

```
#include <rep_each.h>
```

Inheritance diagram for Rcpp::sugar::Rep\_each< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Rep\_each< RTYPE, NA, T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, NA, T > [VEC\\_TYPE](#)
- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)

## Public Member Functions

- [Rep\\_each](#) (const [VEC\\_TYPE](#) &object\_, R\_xlen\_t times\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [VEC\\_TYPE](#) & [object](#)
- [R\\_xlen\\_t times](#)
- [R\\_xlen\\_t n](#)

### 6.669.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Rep_each< RTYPE, NA, T >
```

Definition at line 29 of file rep\_each.h.

### 6.669.2 Member Typedef Documentation

#### 6.669.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Rep\_each< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file rep\_each.h.

#### 6.669.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Rep\_each< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file rep\_each.h.

## 6.669.3 Constructor & Destructor Documentation

### 6.669.3.1 Rep\_each()

```
template<int RTYPE, bool NA, typename T >  
Rcpp::sugar::Rep_each< RTYPE, NA, T >::Rep_each (   
    const VEC_TYPE & object_,  
    R_xlen_t times_ ) [inline]
```

Definition at line 34 of file rep\_each.h.

## 6.669.4 Member Function Documentation

### 6.669.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >  
STORAGE Rcpp::sugar::Rep_each< RTYPE, NA, T >::operator[] (   
    R_xlen_t i ) const [inline]
```

Definition at line 37 of file rep\_each.h.

References Rcpp::sugar::Rep\_each< RTYPE, NA, T >::times.

### 6.669.4.2 size()

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Rep_each< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 40 of file rep\_each.h.

References Rcpp::sugar::Rep\_each< RTYPE, NA, T >::n, and Rcpp::sugar::Rep\_each< RTYPE, NA, T >::times.

## 6.669.5 Member Data Documentation

### 6.669.5.1 n

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Rep_each< RTYPE, NA, T >::n [private]
```

Definition at line 44 of file rep\_each.h.

Referenced by Rcpp::sugar::Rep\_each< RTYPE, NA, T >::size().

### 6.669.5.2 object

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::sugar::Rep_each< RTYPE, NA, T >::object [private]
```

Definition at line 43 of file rep\_each.h.

### 6.669.5.3 times

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Rep_each< RTYPE, NA, T >::times [private]
```

Definition at line 44 of file rep\_each.h.

Referenced by Rcpp::sugar::Rep\_each< RTYPE, NA, T >::operator[](), and Rcpp::sugar::Rep\_each< RTYPE, NA, T >::size().

The documentation for this class was generated from the following file:

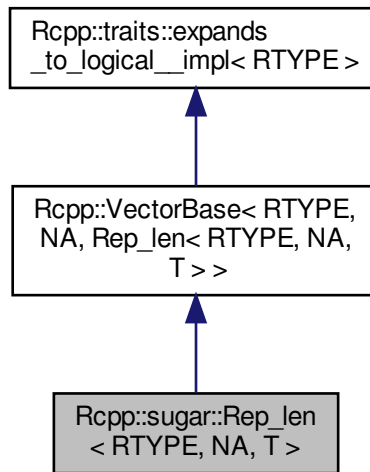
- [inst/include/Rcpp/sugar/functions/rep\\_each.h](#)



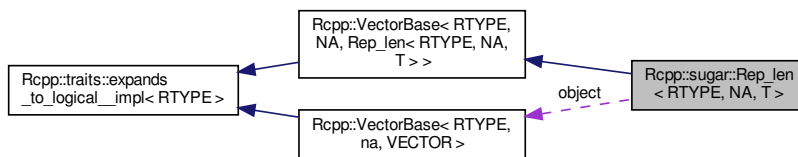
## 6.670 Rcpp::sugar::Rep\_len< RTYPE, NA, T > Class Template Reference

```
#include <rep_len.h>
```

Inheritance diagram for Rcpp::sugar::Rep\_len< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Rep\_len< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`

### Public Member Functions

- `Rep_len` (const `VEC_TYPE` &object\_, `R_xlen_t` len\_)
- `STORAGE` `operator[]` (`R_xlen_t` i) const
- `R_xlen_t` `size` () const

## Private Attributes

- const [VEC\\_TYPE](#) & *object*
- [R\\_xlen\\_t](#) *len*
- [R\\_xlen\\_t](#) *n*

### 6.670.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::Rep_len< RTYPE, NA, T >
```

Definition at line 29 of file `rep_len.h`.

### 6.670.2 Member Typedef Documentation

#### 6.670.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Rep_len< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file `rep_len.h`.

#### 6.670.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Rep_len< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file `rep_len.h`.

### 6.670.3 Constructor & Destructor Documentation

#### 6.670.3.1 Rep\_len()

```
template<int RTYPE, bool NA, typename T >  
Rcpp::sugar::Rep_len< RTYPE, NA, T >::Rep_len (  
    const VEC\_TYPE & object_,  
    R\_xlen\_t len_ ) [inline]
```

Definition at line 34 of file `rep_len.h`.

## 6.670.4 Member Function Documentation

### 6.670.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >  
STORAGE Rcpp::sugar::Rep_len< RTYPE, NA, T >::operator[] (   
    R_xlen_t i ) const [inline]
```

Definition at line 37 of file rep\_len.h.

References Rcpp::sugar::Rep\_len< RTYPE, NA, T >::n.

### 6.670.4.2 size()

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Rep_len< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 40 of file rep\_len.h.

References Rcpp::sugar::Rep\_len< RTYPE, NA, T >::len.

## 6.670.5 Member Data Documentation

### 6.670.5.1 len

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Rep_len< RTYPE, NA, T >::len [private]
```

Definition at line 44 of file rep\_len.h.

Referenced by Rcpp::sugar::Rep\_len< RTYPE, NA, T >::size().

### 6.670.5.2 n

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rep_len< RTYPE, NA, T >::n [private]
```

Definition at line 44 of file rep\_len.h.

Referenced by Rcpp::sugar::Rep\_len< RTYPE, NA, T >::operator[]().

### 6.670.5.3 object

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::Rep_len< RTYPE, NA, T >::object [private]
```

Definition at line 43 of file rep\_len.h.

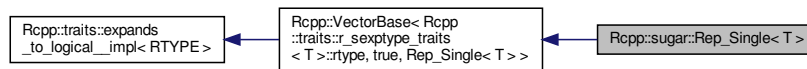
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/rep\\_len.h](#)

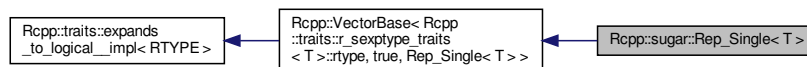
## 6.671 Rcpp::sugar::Rep\_Single< T > Class Template Reference

```
#include <rep.h>
```

Inheritance diagram for Rcpp::sugar::Rep\_Single< T >:



Collaboration diagram for Rcpp::sugar::Rep\_Single< T >:



## Public Member Functions

- [Rep\\_Single](#) (const T &x\_, R\_xlen\_t n\_)
- T [operator\[\]](#) (R\_xlen\_t) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const T & [x](#)
- R\_xlen\_t [n](#)

## Additional Inherited Members

### 6.671.1 Detailed Description

```
template<typename T>  
class Rcpp::sugar::Rep_Single< T >
```

Definition at line 48 of file rep.h.

### 6.671.2 Constructor & Destructor Documentation

#### 6.671.2.1 Rep\_Single()

```
template<typename T >  
Rcpp::sugar::Rep_Single< T >::Rep_Single (  
    const T & x_,  
    R_xlen_t n_ ) [inline]
```

Definition at line 54 of file rep.h.

### 6.671.3 Member Function Documentation

### 6.671.3.1 operator[]()

```
template<typename T >
T Rcpp::sugar::Rep_Single< T >::operator[] (
    R_xlen_t ) const [inline]
```

Definition at line 56 of file rep.h.

References Rcpp::sugar::Rep\_Single< T >::x.

### 6.671.3.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Rep_Single< T >::size ( ) const [inline]
```

Definition at line 59 of file rep.h.

References Rcpp::sugar::Rep\_Single< T >::n.

## 6.671.4 Member Data Documentation

### 6.671.4.1 n

```
template<typename T >
R_xlen_t Rcpp::sugar::Rep_Single< T >::n [private]
```

Definition at line 63 of file rep.h.

Referenced by Rcpp::sugar::Rep\_Single< T >::size().

### 6.671.4.2 x

```
template<typename T >
const T& Rcpp::sugar::Rep_Single< T >::x [private]
```

Definition at line 62 of file rep.h.

Referenced by Rcpp::sugar::Rep\_Single< T >::operator[]().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/rep.h](#)

## 6.672 Rcpp::result< T > Class Template Reference

```
#include <Module.h>
```

### Public Member Functions

- [result](#) (T \*ptr\_)
- [operator T\\*](#) ()

### Private Attributes

- T \* [ptr](#)

### 6.672.1 Detailed Description

```
template<typename T>  
class Rcpp::result< T >
```

Definition at line 56 of file Module.h.

### 6.672.2 Constructor & Destructor Documentation

#### 6.672.2.1 result()

```
template<typename T >  
Rcpp::result< T >::result (  
    T * ptr_ ) [inline]
```

Definition at line 58 of file Module.h.

### 6.672.3 Member Function Documentation

#### 6.672.3.1 operator T\*()

```
template<typename T >  
Rcpp::result< T >::operator T* ( ) [inline]
```

Definition at line 59 of file Module.h.

References [Rcpp::result< T >::ptr](#).

## 6.672.4 Member Data Documentation

### 6.672.4.1 ptr

```
template<typename T >
T* Rcpp::result< T >::ptr [private]
```

Definition at line 61 of file Module.h.

Referenced by Rcpp::result< T >::operator T\*().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

## 6.673 Rcpp::sugar::median\_detail::result< RTYPE > Struct Template Reference

```
#include <median.h>
```

### Public Types

- enum { rtype = RTYPE }
- typedef Rcpp::traits::storage\_type< RTYPE >::type type

### 6.673.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::sugar::median_detail::result< RTYPE >
```

Definition at line 35 of file median.h.

### 6.673.2 Member Typedef Documentation



### 6.673.2.1 type

```
template<int RTYPE>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::median_detail::result< RTYPE >↔
::type
```

Definition at line 36 of file median.h.

## 6.673.3 Member Enumeration Documentation

### 6.673.3.1 anonymous enum

```
template<int RTYPE>
anonymous enum
```

#### Enumerator

rtype	
-------	--

Definition at line 37 of file median.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/median.h](#)

## 6.674 Rcpp::sugar::median\_detail::result< INTSXP > Struct Reference

```
#include <median.h>
```

### Public Types

- enum { [rtype](#) = REALSXP }
- typedef double [type](#)

### 6.674.1 Detailed Description

Definition at line 41 of file median.h.

## 6.674.2 Member Typedef Documentation

### 6.674.2.1 type

```
typedef double Rcpp::sugar::median_detail::result< INTSXP >::type
```

Definition at line 42 of file median.h.

## 6.674.3 Member Enumeration Documentation

### 6.674.3.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	
-------	--

Definition at line 43 of file median.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/median.h](#)

## 6.675 Rcpp::sugar::median\_detail::result< STRSXP > Struct Reference

```
#include <median.h>
```

### Public Types

- enum { [rtype](#) = STRSXP }
- typedef [Rcpp::String](#) [type](#)

### 6.675.1 Detailed Description

Definition at line 47 of file median.h.

## 6.675.2 Member Typedef Documentation

### 6.675.2.1 type

```
typedef Rcpp::String Rcpp::sugar::median_detail::result< STRSXP >::type
```

Definition at line 48 of file median.h.

## 6.675.3 Member Enumeration Documentation

### 6.675.3.1 anonymous enum

anonymous enum

Enumerator

rtype	
-------	--

Definition at line 49 of file median.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/[median.h](#)

## 6.676 Rcpp::traits::result\_of< T > Struct Template Reference

```
#include <result_of.h>
```

### Public Types

- typedef T::result\_type [type](#)

### 6.676.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::result_of< T >
```

Definition at line 30 of file result\_of.h.

## 6.676.2 Member Typedef Documentation

### 6.676.2.1 type

```
template<typename T >  
typedef T::result_type Rcpp::traits::result_of< T >::type
```

Definition at line 31 of file result\_of.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/result\_of.h

## 6.677 Rcpp::traits::result\_of< RESULT\_TYPE(\*) (INPUT\_TYPE) > Struct Template Reference

```
#include <result_of.h>
```

### Public Types

- typedef RESULT\_TYPE [type](#)

### 6.677.1 Detailed Description

```
template<typename RESULT_TYPE, typename INPUT_TYPE>  
struct Rcpp::traits::result_of< RESULT_TYPE(*) (INPUT_TYPE) >
```

Definition at line 35 of file result\_of.h.

## 6.677.2 Member Typedef Documentation

### 6.677.2.1 type

```
template<typename RESULT_TYPE , typename INPUT_TYPE >  
typedef RESULT_TYPE Rcpp::traits::result_of< RESULT_TYPE(*) (INPUT_TYPE) >::type
```

Definition at line 36 of file result\_of.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/result\_of.h

## 6.678 Rcpp::traits::result\_of< RESULT\_TYPE(\*) (U1, U2) > Struct Template Reference

```
#include <result_of.h>
```

### Public Types

- typedef RESULT\_TYPE [type](#)

### 6.678.1 Detailed Description

```
template<typename RESULT_TYPE, typename U1, typename U2>  
struct Rcpp::traits::result_of< RESULT_TYPE(*) (U1, U2) >
```

Definition at line 40 of file result\_of.h.

### 6.678.2 Member Typedef Documentation

#### 6.678.2.1 type

```
template<typename RESULT_TYPE , typename U1 , typename U2 >  
typedef RESULT_TYPE Rcpp::traits::result\_of< RESULT\_TYPE\(\*\) \(U1, U2\) >::type
```

Definition at line 41 of file result\_of.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[result\\_of.h](#)

## 6.679 Rcpp::traits::result\_of< RESULT\_TYPE(\*) (U1, U2, U3) > Struct Template Reference

```
#include <result_of.h>
```

### Public Types

- typedef RESULT\_TYPE [type](#)

### 6.679.1 Detailed Description

```
template<typename RESULT_TYPE, typename U1, typename U2, typename U3>
struct Rcpp::traits::result_of< RESULT_TYPE*(U1, U2, U3) >
```

Definition at line 45 of file result\_of.h.

### 6.679.2 Member Typedef Documentation

#### 6.679.2.1 type

```
template<typename RESULT_TYPE , typename U1 , typename U2 , typename U3 >
typedef RESULT_TYPE Rcpp::traits::result_of< RESULT_TYPE*(U1, U2, U3) >::type
```

Definition at line 46 of file result\_of.h.

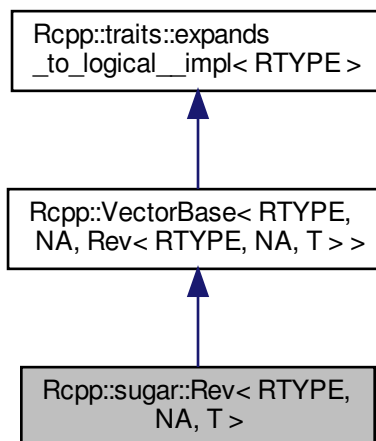
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/result\\_of.h](#)

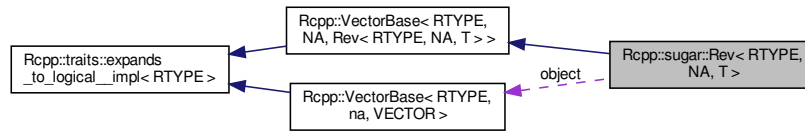
## 6.680 Rcpp::sugar::Rev< RTYPE, NA, T > Class Template Reference

```
#include <rev.h>
```

Inheritance diagram for Rcpp::sugar::Rev< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Rev< RTYPE, NA, T >:



## Public Types

- typedef [Rcpp::VectorBase< RTYPE, NA, T > VEC\\_TYPE](#)
- typedef [Rcpp::traits::storage\\_type< RTYPE >::type STORAGE](#)

## Public Member Functions

- [Rev](#) (const [VEC\\_TYPE](#) &object\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [VEC\\_TYPE](#) & [object](#)
- [R\\_xlen\\_t n](#)

### 6.680.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Rev< RTYPE, NA, T >
```

Definition at line 29 of file rev.h.

### 6.680.2 Member Typedef Documentation

#### 6.680.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Rev< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file rev.h.

### 6.680.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Rev< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file rev.h.

## 6.680.3 Constructor & Destructor Documentation

### 6.680.3.1 Rev()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Rev< RTYPE, NA, T >::Rev (
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 34 of file rev.h.

## 6.680.4 Member Function Documentation

### 6.680.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Rev< RTYPE, NA, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 37 of file rev.h.

References `Rcpp::sugar::Rev< RTYPE, NA, T >::n`.

### 6.680.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Rev< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 40 of file rev.h.

References `Rcpp::sugar::Rev< RTYPE, NA, T >::n`.



## 6.680.5 Member Data Documentation

### 6.680.5.1 n

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Rev< RTYPE, NA, T >::n [private]
```

Definition at line 44 of file rev.h.

Referenced by `Rcpp::sugar::Rev< RTYPE, NA, T >::operator[]()`, and `Rcpp::sugar::Rev< RTYPE, NA, T >::size()`.

### 6.680.5.2 object

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::sugar::Rev< RTYPE, NA, T >::object [private]
```

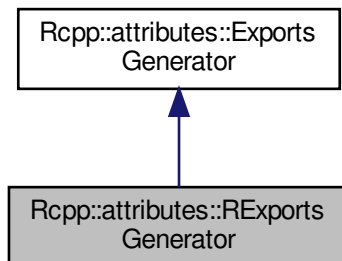
Definition at line 43 of file rev.h.

The documentation for this class was generated from the following file:

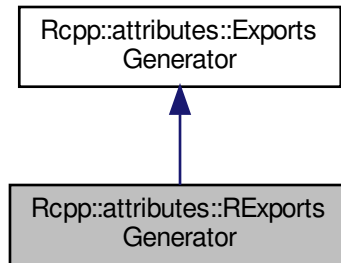
- `inst/include/Rcpp/sugar/functions/rev.h`

## 6.681 Rcpp::attributes::REExportsGenerator Class Reference

Inheritance diagram for `Rcpp::attributes::REExportsGenerator`:



Collaboration diagram for Rcpp::attributes::REExportsGenerator:



## Public Member Functions

- [REExportsGenerator](#) (const std::string &packageDir, const std::string &package, bool registration, const std::string &fileSep)
- virtual void [writeBegin](#) ()
- virtual void [writeEnd](#) (bool hasPackageInit)
- virtual bool [commit](#) (const std::vector< std::string > &includes)

## Private Member Functions

- virtual void [doWriteFunctions](#) (const [SourceFileAttributes](#) &attributes, bool verbose)

## Private Attributes

- bool [registration\\_](#)

## Additional Inherited Members

### 6.681.1 Detailed Description

Definition at line 770 of file attributes.cpp.

### 6.681.2 Constructor & Destructor Documentation

### 6.681.2.1 RExportsGenerator()

```
Rcpp::attributes::REExportsGenerator::REExportsGenerator (
    const std::string & packageDir,
    const std::string & package,
    bool registration,
    const std::string & fileSep )
```

Definition at line 2449 of file attributes.cpp.

## 6.681.3 Member Function Documentation

### 6.681.3.1 commit()

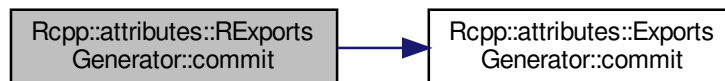
```
bool Rcpp::attributes::REExportsGenerator::commit (
    const std::vector< std::string > & includes ) [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 2549 of file attributes.cpp.

References [Rcpp::attributes::ExportsGenerator::commit\(\)](#).

Here is the call graph for this function:



### 6.681.3.2 doWriteFunctions()

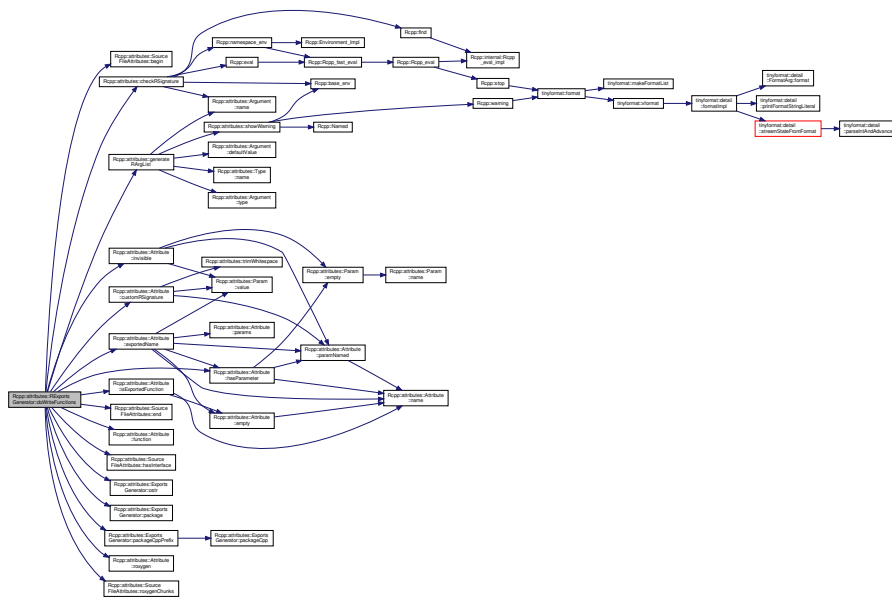
```
void Rcpp::attributes::REExportsGenerator::doWriteFunctions (
    const SourceFileAttributes & attributes,
    bool verbose ) [private], [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 2461 of file attributes.cpp.

References [Rcpp::attributes::SourceFileAttributes::begin\(\)](#), [Rcpp::attributes::checkRSignature\(\)](#), [Rcpp::attributes::Attribute::customRSignature\(\)](#), [Rcpp::attributes::SourceFileAttributes::end\(\)](#), [Rcpp::attributes::Attribute::exportedName\(\)](#), [Rcpp::attributes::Attribute::function\(\)](#), [Rcpp::attributes::generateRArgList\(\)](#), [Rcpp::attributes::SourceFileAttributes::hasInterface\(\)](#), [Rcpp::attributes::Attribute::hasParameter\(\)](#), [Rcpp::attributes::Attribute::invisible\(\)](#), [Rcpp::attributes::Attribute::isExportedFunction\(\)](#), [Rcpp::attributes::kExportSignature](#), [Rcpp::attributes::kInterfaceR](#), [Rcpp::attributes::ExportsGenerator::ostr\(\)](#), [Rcpp::attributes::ExportsGenerator::package\(\)](#), [Rcpp::attributes::ExportsGenerator::packageCppPrefix\(\)](#), [registration\\_](#), [Rcpp::attributes::Attribute::roxygen\(\)](#), and [Rcpp::attributes::SourceFileAttributes::roxygenChunks\(\)](#).

Here is the call graph for this function:



### 6.681.3.3 writeBegin()

```
virtual void Rcpp::attributes::REExportsGenerator::writeBegin ( ) [inline], [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 777 of file attributes.cpp.

### 6.681.3.4 writeEnd()

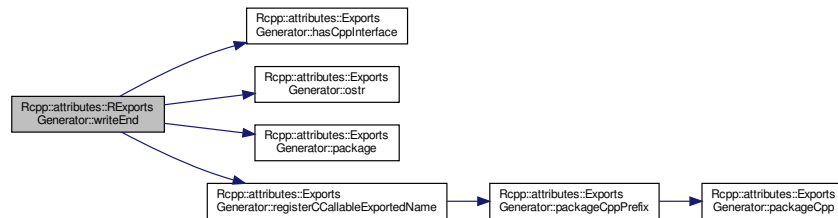
```
void Rcpp::attributes::REExportsGenerator::writeEnd (
    bool hasPackageInit ) [virtual]
```

Implements [Rcpp::attributes::ExportsGenerator](#).

Definition at line 2537 of file attributes.cpp.

References [Rcpp::attributes::ExportsGenerator::hasCppInterface\(\)](#), [Rcpp::attributes::ExportsGenerator::ostr\(\)](#), [Rcpp::attributes::ExportsGenerator::package\(\)](#), and [Rcpp::attributes::ExportsGenerator::registerCCallableExportedName\(\)](#).

Here is the call graph for this function:



## 6.681.4 Member Data Documentation

### 6.681.4.1 registration\_

```
bool Rcpp::attributes::REExportsGenerator::registration_ [private]
```

Definition at line 785 of file attributes.cpp.

Referenced by `doWriteFunctions()`.

The documentation for this class was generated from the following file:

- [src/attributes.cpp](#)

## 6.682 Rcpp::RNGScope Class Reference

```
#include <RNGScope.h>
```

## Public Member Functions

- [RNGScope \(\)](#)
- [~RNGScope \(\)](#)

### 6.682.1 Detailed Description

Definition at line 27 of file RNGScope.h.

### 6.682.2 Constructor & Destructor Documentation

#### 6.682.2.1 RNGScope()

```
Rcpp::RNGScope::RNGScope ( ) [inline]
```

Definition at line 29 of file RNGScope.h.

References [Rcpp::internal::enterRNGScope\(\)](#).

Here is the call graph for this function:



#### 6.682.2.2 ~RNGScope()

```
Rcpp::RNGScope::~~RNGScope ( ) [inline]
```

Definition at line 30 of file RNGScope.h.

References [Rcpp::internal::exitRNGScope\(\)](#).

Here is the call graph for this function:



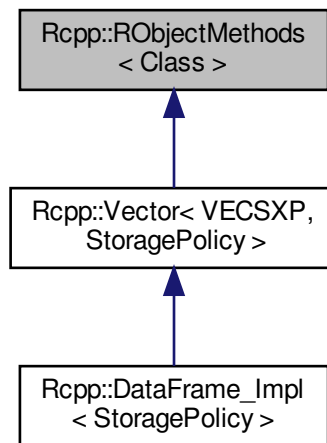
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/RNGScope.h](#)

## 6.683 Rcpp::RObjectMethods< Class > Class Template Reference

```
#include <RObjectMethods.h>
```

Inheritance diagram for Rcpp::RObjectMethods< Class >:



### Public Member Functions

- bool `isNULL` () const
- int `sexp_type` () const
- bool `isObject` () const
- bool `isS4` () const

### 6.683.1 Detailed Description

```
template<typename Class>  
class Rcpp::RObjectMethods< Class >
```

Definition at line 24 of file `RObjectMethods.h`.

### 6.683.2 Member Function Documentation

### 6.683.2.1 isNULL()

```
template<typename Class >  
bool Rcpp::RObjectMethods< Class >::isNULL ( ) const [inline]
```

Definition at line 27 of file RObjectMethods.h.

### 6.683.2.2 isObject()

```
template<typename Class >  
bool Rcpp::RObjectMethods< Class >::isObject ( ) const [inline]
```

Definition at line 35 of file RObjectMethods.h.

### 6.683.2.3 isS4()

```
template<typename Class >  
bool Rcpp::RObjectMethods< Class >::isS4 ( ) const [inline]
```

Definition at line 39 of file RObjectMethods.h.

### 6.683.2.4 sexp\_type()

```
template<typename Class >  
int Rcpp::RObjectMethods< Class >::sexp_type ( ) const [inline]
```

Definition at line 31 of file RObjectMethods.h.

The documentation for this class was generated from the following file:

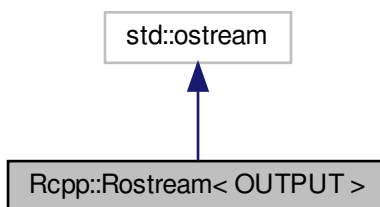
- [inst/include/Rcpp/proxy/RObjectMethods.h](#)



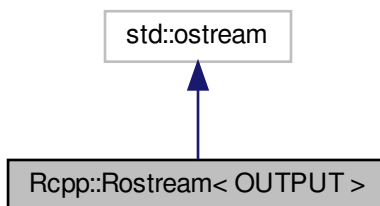
## 6.684 Rcpp::Rostream< OUTPUT > Class Template Reference

```
#include <Rstreambuf.h>
```

Inheritance diagram for Rcpp::Rostream< OUTPUT >:



Collaboration diagram for Rcpp::Rostream< OUTPUT >:



### Public Member Functions

- [Rostream\(\)](#)

### Private Types

- typedef [Rstreambuf< OUTPUT >](#) [Buffer](#)

### Private Attributes

- [Buffer](#) [buf](#)

### 6.684.1 Detailed Description

```
template<bool OUTPUT>
class Rcpp::Rostream< OUTPUT >
```

Definition at line 45 of file Rstreambuf.h.

### 6.684.2 Member Typedef Documentation

#### 6.684.2.1 Buffer

```
template<bool OUTPUT>
typedef Rstreambuf<OUTPUT> Rcpp::Rostream< OUTPUT >::Buffer [private]
```

Definition at line 46 of file Rstreambuf.h.

### 6.684.3 Constructor & Destructor Documentation

#### 6.684.3.1 Rostream()

```
template<bool OUTPUT>
Rcpp::Rostream< OUTPUT >::Rostream ( ) [inline]
```

Definition at line 49 of file Rstreambuf.h.

### 6.684.4 Member Data Documentation

#### 6.684.4.1 buf

```
template<bool OUTPUT>
Buffer Rcpp::Rostream< OUTPUT >::buf [private]
```

Definition at line 47 of file Rstreambuf.h.

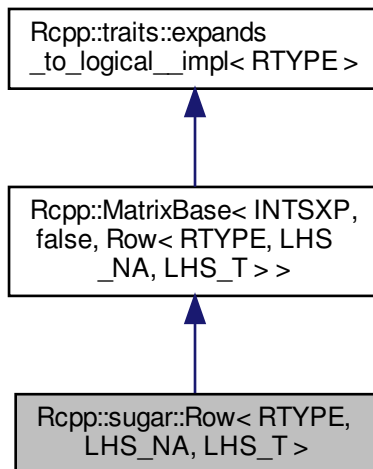
The documentation for this class was generated from the following file:

- inst/include/Rcpp/iostream/[Rstreambuf.h](#)

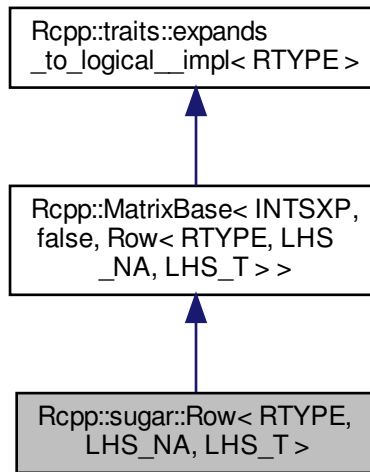
## 6.685 Rcpp::sugar::Row< RTYPE, LHS\_NA, LHS\_T > Class Template Reference

```
#include <row.h>
```

Inheritance diagram for Rcpp::sugar::Row< RTYPE, LHS\_NA, LHS\_T >:



Collaboration diagram for `Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >`:



## Public Types

- typedef `Rcpp::MatrixBase< RTYPE, LHS_NA, LHS_T >` `LHS_TYPE`

## Public Member Functions

- `Row` (const `LHS_TYPE` &lhs)
- `operator()` (int i, int j) const
- `R_xlen_t size` () const
- `int nrow` () const
- `int ncol` () const

## Private Attributes

- `int nr`
- `int nc`

## Additional Inherited Members

### 6.685.1 Detailed Description

```

template<int RTYPE, bool LHS_NA, typename LHS_T>
class Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >
  
```

Definition at line 29 of file row.h.

## 6.685.2 Member Typedef Documentation

### 6.685.2.1 LHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
typedef Rcpp::MatrixBase<RTYPE,LHS_NA,LHS_T> Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::LHS_TYPE
```

Definition at line 35 of file row.h.

## 6.685.3 Constructor & Destructor Documentation

### 6.685.3.1 Row()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::Row (
    const LHS_TYPE & lhs ) [inline]
```

Definition at line 37 of file row.h.

## 6.685.4 Member Function Documentation

### 6.685.4.1 ncol()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::ncol ( ) const [inline]
```

Definition at line 45 of file row.h.

References Rcpp::sugar::Row< RTYPE, LHS\_NA, LHS\_T >::nc.

### 6.685.4.2 nrow()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::nrow ( ) const [inline]
```

Definition at line 44 of file row.h.

References Rcpp::sugar::Row< RTYPE, LHS\_NA, LHS\_T >::nr.

### 6.685.4.3 operator()()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::operator() (
    int i,
    int j ) const [inline]
```

Definition at line 39 of file row.h.

### 6.685.4.4 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
R_xlen_t Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::size ( ) const [inline]
```

Definition at line 43 of file row.h.

References Rcpp::sugar::Row< RTYPE, LHS\_NA, LHS\_T >::nc, and Rcpp::sugar::Row< RTYPE, LHS\_NA, LHS\_T >::nr.

## 6.685.5 Member Data Documentation

### 6.685.5.1 nc

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::nc [private]
```

Definition at line 48 of file row.h.

Referenced by Rcpp::sugar::Row< RTYPE, LHS\_NA, LHS\_T >::ncol(), and Rcpp::sugar::Row< RTYPE, LHS\_NA, LHS\_T >::size().

## 6.685.5.2 nr

```
template<int RTYPE, bool LHS_NA, typename LHS_T >
int Rcpp::sugar::Row< RTYPE, LHS_NA, LHS_T >::nr [private]
```

Definition at line 48 of file row.h.

Referenced by Rcpp::sugar::Row< RTYPE, LHS\_NA, LHS\_T >::nrow(), and Rcpp::sugar::Row< RTYPE, LHS\_NA, LHS\_T >::size().

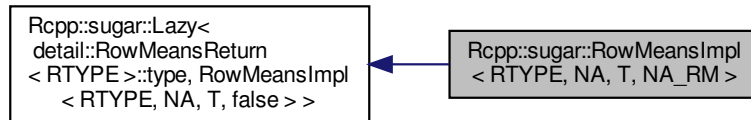
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/matrix/row.h

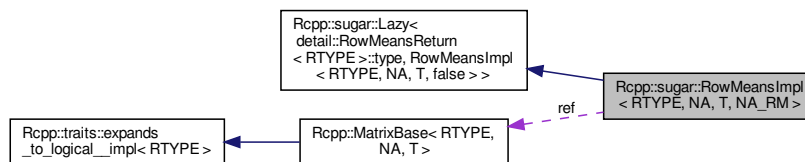
## 6.686 Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA\_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA\_RM >:



Collaboration diagram for Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA\_RM >:



### Public Member Functions

- [RowMeansImpl](#) (const [MatrixBase](#)< RTYPE, NA, T > &ref\_)
- [return\\_vector get](#) () const

## Private Types

- typedef [detail::RowMeansReturn](#)< RTYPE > [return\\_traits](#)
- typedef [return\\_traits::type](#) [return\\_vector](#)
- typedef [traits::storage\\_type](#)< [return\\_traits::rtype](#) >::type [stored\\_type](#)

## Private Attributes

- const [MatrixBase](#)< RTYPE, NA, T > & [ref](#)

### 6.686.1 Detailed Description

```
template<int RTYPE, bool NA, typename T, bool NA_RM = false>
class Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >
```

Definition at line 507 of file rowSums.h.

### 6.686.2 Member Typedef Documentation

#### 6.686.2.1 return\_traits

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef detail::RowMeansReturn<RTYPE> Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::return\_traits
[private]
```

Definition at line 512 of file rowSums.h.

#### 6.686.2.2 return\_vector

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef return\_traits::type Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::return\_vector [private]
```

Definition at line 513 of file rowSums.h.



### 6.686.2.3 stored\_type

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::RowMeansImpl< RTYPE, NA, T,
NA_RM >::stored_type [private]
```

Definition at line 514 of file rowSums.h.

## 6.686.3 Constructor & Destructor Documentation

### 6.686.3.1 RowMeansImpl()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::RowMeansImpl (
    const MatrixBase< RTYPE, NA, T > & ref_ ) [inline]
```

Definition at line 517 of file rowSums.h.

## 6.686.4 Member Function Documentation

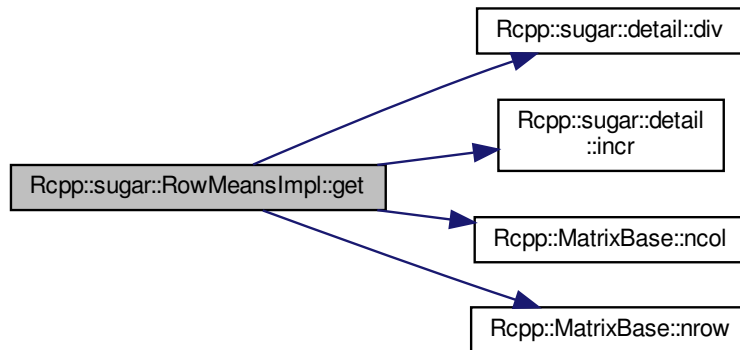
### 6.686.4.1 get()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
return_vector Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::get ( ) const [inline]
```

Definition at line 521 of file rowSums.h.

References [Rcpp::sugar::detail::div\(\)](#), [Rcpp::sugar::detail::incr\(\)](#), [Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol\(\)](#), [Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow\(\)](#), and [Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA\\_RM >::ref](#).

Here is the call graph for this function:



## 6.686.5 Member Data Documentation

### 6.686.5.1 ref

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::ref [private]
```

Definition at line 510 of file rowSums.h.

Referenced by `Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::get()`, and `Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get()`.

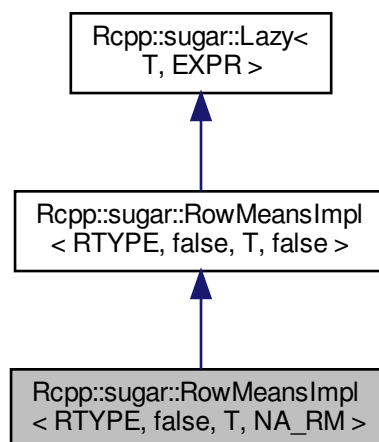
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/rowSums.h`

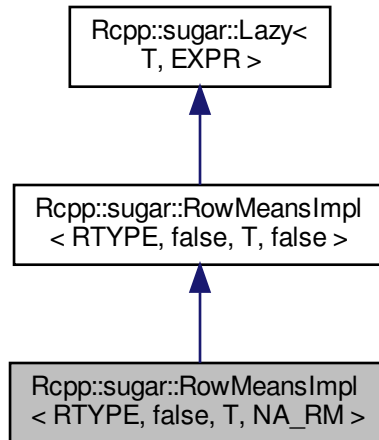
## 6.687 Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA\_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for `Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA_RM >`:



Collaboration diagram for Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA\_RM >:



## Additional Inherited Members

### 6.687.1 Detailed Description

```
template<int RTYPE, typename T, bool NA_RM>
class Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA_RM >
```

Definition at line 706 of file rowSums.h.

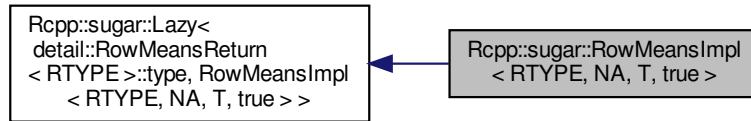
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/rowSums.h](#)

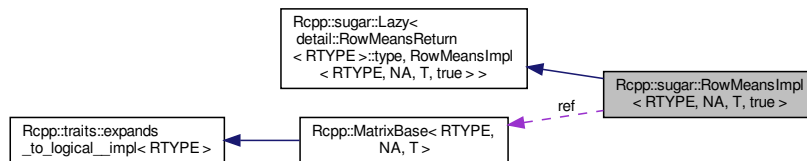
## 6.688 Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for `Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >`:



Collaboration diagram for `Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >`:



## Public Member Functions

- [RowMeansImpl](#) (const [MatrixBase](#)< RTYPE, NA, T > &ref\_)
- [return\\_vector](#) [get](#) () const

## Private Types

- typedef [detail::RowMeansReturn](#)< RTYPE > [return\\_traits](#)
- typedef [return\\_traits::type](#) [return\\_vector](#)
- typedef [traits::storage\\_type](#)< [return\\_traits::rtype](#) >::type [stored\\_type](#)

## Private Attributes

- const [MatrixBase](#)< RTYPE, NA, T > & [ref](#)

## 6.688.1 Detailed Description

```

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >
  
```

Definition at line 603 of file `rowSums.h`.

## 6.688.2 Member Typedef Documentation

### 6.688.2.1 return\_traits

```
template<int RTYPE, bool NA, typename T >
typedef detail::RowMeansReturn<RTYPE> Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::return_traits
[private]
```

Definition at line 608 of file rowSums.h.

### 6.688.2.2 return\_vector

```
template<int RTYPE, bool NA, typename T >
typedef return_traits::type Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::return_vector [private]
```

Definition at line 609 of file rowSums.h.

### 6.688.2.3 stored\_type

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::RowMeansImpl< RTYPE, NA, T,
true >::stored_type [private]
```

Definition at line 610 of file rowSums.h.

## 6.688.3 Constructor & Destructor Documentation

### 6.688.3.1 RowMeansImpl()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::RowMeansImpl (
    const MatrixBase< RTYPE, NA, T > & ref_ ) [inline]
```

Definition at line 613 of file rowSums.h.

## 6.688.4 Member Function Documentation

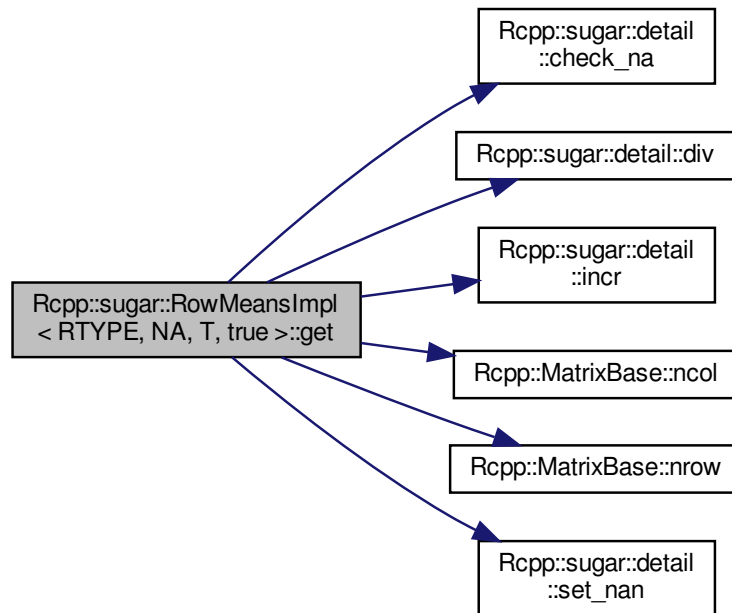
### 6.688.4.1 get()

```
template<int RTYPE, bool NA, typename T >
return_vector Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::get ( ) const [inline]
```

Definition at line 617 of file rowSums.h.

References `Rcpp::sugar::detail::check_na()`, `Rcpp::sugar::detail::div()`, `Rcpp::sugar::detail::incr()`, `Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol()`, `Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow()`, `Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA_RM >::ref`, and `Rcpp::sugar::detail::set_nan()`.

Here is the call graph for this function:



## 6.688.5 Member Data Documentation

## 6.688.5.1 ref

```
template<int RTYPE, bool NA, typename T >
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >::ref [private]
```

Definition at line 606 of file rowSums.h.

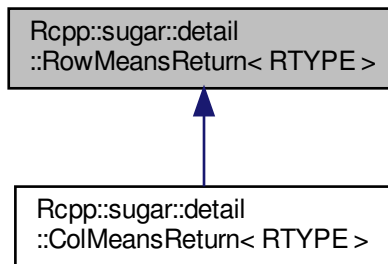
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/rowSums.h

## 6.689 Rcpp::sugar::detail::RowMeansReturn< RTYPE > Struct Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::detail::RowMeansReturn< RTYPE >:



### Public Types

- enum { `rtype` = REALSXP }
- typedef `Vector`< REALSXP > `type`

### 6.689.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::sugar::detail::RowMeansReturn< RTYPE >
```

Definition at line 103 of file rowSums.h.

## 6.689.2 Member Typedef Documentation

### 6.689.2.1 type

```
template<int RTYPE>
typedef Vector<REALSXP> Rcpp::sugar::detail::RowMeansReturn< RTYPE >::type
```

Definition at line 104 of file rowSums.h.

## 6.689.3 Member Enumeration Documentation

### 6.689.3.1 anonymous enum

```
template<int RTYPE>
anonymous enum
```

#### Enumerator

rtype	
-------	--

Definition at line 105 of file rowSums.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/[rowSums.h](#)

## 6.690 Rcpp::sugar::detail::RowMeansReturn< CPLXSXP > Struct Reference

```
#include <rowSums.h>
```

### Public Types

- enum { [rtype](#) = CPLXSXP }
- typedef [Vector](#)< CPLXSXP > [type](#)



## 6.690.1 Detailed Description

Definition at line 109 of file rowSums.h.

## 6.690.2 Member Typedef Documentation

### 6.690.2.1 type

```
typedef Vector<CPLXSP> Rcpp::sugar::detail::RowMeansReturn< CPLXSP >::type
```

Definition at line 110 of file rowSums.h.

## 6.690.3 Member Enumeration Documentation

### 6.690.3.1 anonymous enum

```
anonymous enum
```

Enumerator

rtype	
-------	--

Definition at line 111 of file rowSums.h.

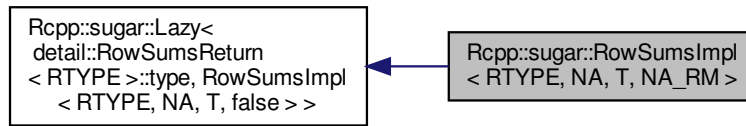
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/rowSums.h](#)

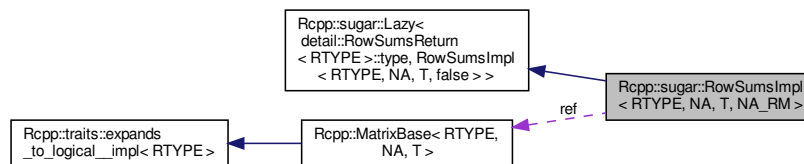
## 6.691 Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA\_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for `Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >`:



Collaboration diagram for `Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >`:



## Public Member Functions

- `RowSumsImpl` (const `MatrixBase< RTYPE, NA, T >` &ref\_)
- `return_vector get` () const

## Private Types

- typedef `detail::RowSumsReturn< RTYPE >` `return_traits`
- typedef `return_traits::type` `return_vector`
- typedef `traits::storage_type< return_traits::rtype >::type` `stored_type`

## Private Attributes

- const `MatrixBase< RTYPE, NA, T >` &ref

### 6.691.1 Detailed Description

```

template<int RTYPE, bool NA, typename T, bool NA_RM = false>
class Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >
  
```

Definition at line 128 of file `rowSums.h`.

## 6.691.2 Member Typedef Documentation

### 6.691.2.1 return\_traits

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef detail::RowSumsReturn<RTYPE> Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::return_traits
[private]
```

Definition at line 133 of file rowSums.h.

### 6.691.2.2 return\_vector

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef return_traits::type Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::return_vector [private]
```

Definition at line 134 of file rowSums.h.

### 6.691.2.3 stored\_type

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::RowSumsImpl< RTYPE, NA, T,
NA_RM >::stored_type [private]
```

Definition at line 135 of file rowSums.h.

## 6.691.3 Constructor & Destructor Documentation

### 6.691.3.1 RowSumsImpl()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::RowSumsImpl (
    const MatrixBase< RTYPE, NA, T > & ref_ ) [inline]
```

Definition at line 138 of file rowSums.h.

## 6.691.4 Member Function Documentation

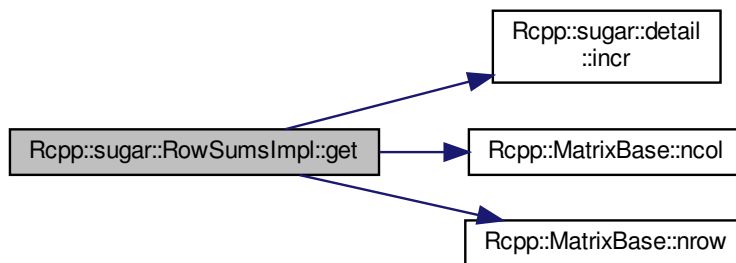
### 6.691.4.1 get()

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
return_vector Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::get ( ) const [inline]
```

Definition at line 142 of file rowSums.h.

References `Rcpp::sugar::detail::incr()`, `Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol()`, `Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow()`, and `Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::ref`.

Here is the call graph for this function:



## 6.691.5 Member Data Documentation

### 6.691.5.1 ref

```
template<int RTYPE, bool NA, typename T , bool NA_RM = false>
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::ref [private]
```

Definition at line 131 of file rowSums.h.

Referenced by `Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA_RM >::get()`, and `Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get()`.

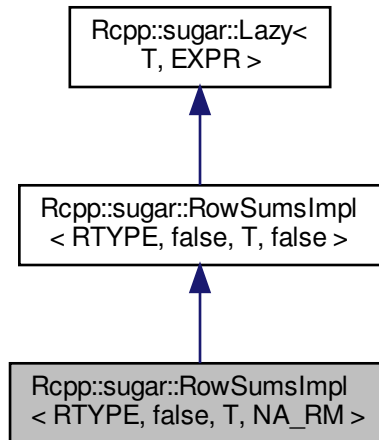
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/rowSums.h`

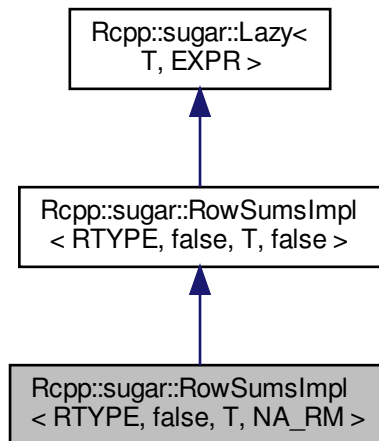
## 6.692 Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA\_RM > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA\_RM >:



Collaboration diagram for Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA\_RM >:



## Additional Inherited Members

### 6.692.1 Detailed Description

```
template<int RTYPE, typename T, bool NA_RM>
class Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA_RM >
```

Definition at line 310 of file rowSums.h.

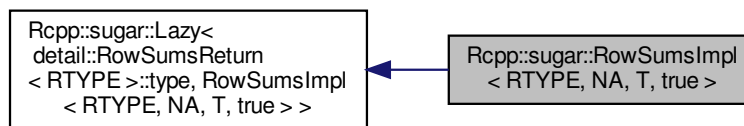
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[rowSums.h](#)

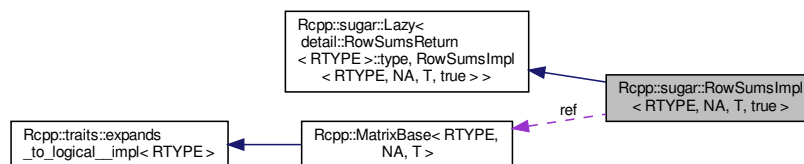
### 6.693 Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true > Class Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >:



Collaboration diagram for Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >:



## Public Member Functions

- [RowSumsImpl](#) (const [MatrixBase](#)< RTYPE, NA, T > &ref\_)
- [return\\_vector get](#) () const

## Private Types

- typedef [detail::RowSumsReturn< RTYPE >](#) [return\\_traits](#)
- typedef [return\\_traits::type](#) [return\\_vector](#)
- typedef [traits::storage\\_type< return\\_traits::rtype >::type](#) [stored\\_type](#)

## Private Attributes

- const [MatrixBase< RTYPE, NA, T >](#) & [ref](#)

### 6.693.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >
```

Definition at line 226 of file rowSums.h.

### 6.693.2 Member Typedef Documentation

#### 6.693.2.1 return\_traits

```
template<int RTYPE, bool NA, typename T >
typedef detail::RowSumsReturn<RTYPE> Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::return\_traits
[private]
```

Definition at line 231 of file rowSums.h.

#### 6.693.2.2 return\_vector

```
template<int RTYPE, bool NA, typename T >
typedef return\_traits::type Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::return\_vector [private]
```

Definition at line 232 of file rowSums.h.

### 6.693.2.3 stored\_type

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<return_traits::rtype>::type Rcpp::sugar::RowSumsImpl< RTYPE, NA, T,
true >::stored_type [private]
```

Definition at line 233 of file rowSums.h.

## 6.693.3 Constructor & Destructor Documentation

### 6.693.3.1 RowSumsImpl()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::RowSumsImpl (
    const MatrixBase< RTYPE, NA, T > & ref_ ) [inline]
```

Definition at line 236 of file rowSums.h.

## 6.693.4 Member Function Documentation

### 6.693.4.1 get()

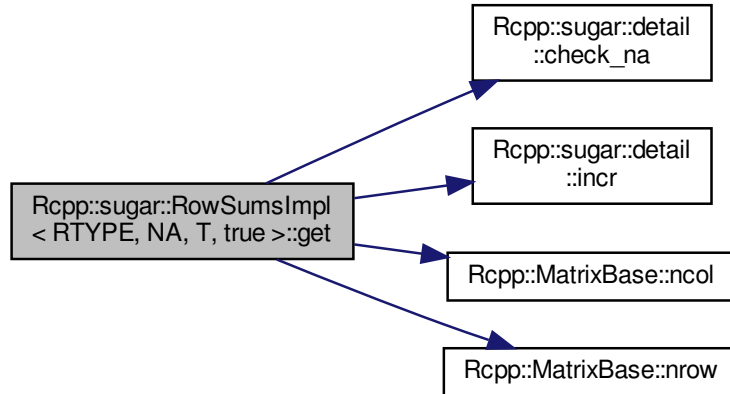
```
template<int RTYPE, bool NA, typename T >
return_vector Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::get ( ) const [inline]
```

Definition at line 240 of file rowSums.h.

References [Rcpp::sugar::detail::check\\_na\(\)](#), [Rcpp::sugar::detail::incr\(\)](#), [Rcpp::MatrixBase< RTYPE, na, MATRIX >::ncol\(\)](#), [Rcpp::MatrixBase< RTYPE, na, MATRIX >::nrow\(\)](#), and [Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA\\_RM >::ref](#).



Here is the call graph for this function:



## 6.693.5 Member Data Documentation

### 6.693.5.1 ref

```
template<int RTYPE, bool NA, typename T >
const MatrixBase<RTYPE, NA, T>& Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >::ref [private]
```

Definition at line 229 of file rowSums.h.

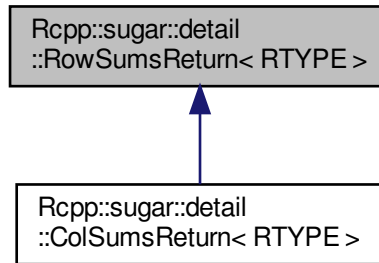
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[rowSums.h](#)

## 6.694 Rcpp::sugar::detail::RowSumsReturn< RTYPE > Struct Template Reference

```
#include <rowSums.h>
```

Inheritance diagram for `Rcpp::sugar::detail::RowSumsReturn< RTYPE >`:



## Public Types

- enum { `rtype` = `RTYPE` }
- typedef `Vector< RTYPE > type`

### 6.694.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::sugar::detail::RowSumsReturn< RTYPE >
```

Definition at line 86 of file `rowSums.h`.

### 6.694.2 Member Typedef Documentation

#### 6.694.2.1 `type`

```
template<int RTYPE>
typedef Vector<RTYPE> Rcpp::sugar::detail::RowSumsReturn< RTYPE >::type
```

Definition at line 87 of file `rowSums.h`.

### 6.694.3 Member Enumeration Documentation

#### 6.694.3.1 anonymous enum

```
template<int RTYPE>
anonymous enum
```

Enumerator

rtype	
-------	--

Definition at line 88 of file rowSums.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/[rowSums.h](#)

## 6.695 Rcpp::sugar::detail::RowSumsReturn< LGLSXP > Struct Reference

```
#include <rowSums.h>
```

### Public Types

- enum { [rtype](#) = INTSXP }
- typedef [Vector](#)< INTSXP > [type](#)

### 6.695.1 Detailed Description

Definition at line 92 of file rowSums.h.

### 6.695.2 Member Typedef Documentation

#### 6.695.2.1 type

```
typedef Vector<INTSXP> Rcpp::sugar::detail::RowSumsReturn< LGLSXP >::type
```

Definition at line 93 of file rowSums.h.

### 6.695.3 Member Enumeration Documentation

#### 6.695.3.1 anonymous enum

```
anonymous enum
```

## Enumerator

rtype	
-------	--

Definition at line 94 of file rowSums.h.

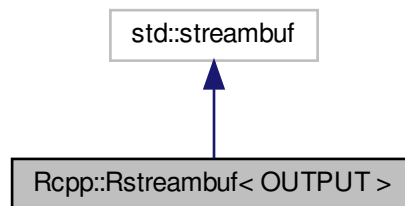
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/rowSums.h](#)

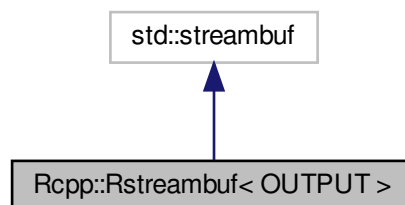
## 6.696 Rcpp::Rstreambuf< OUTPUT > Class Template Reference

```
#include <Rstreambuf.h>
```

Inheritance diagram for Rcpp::Rstreambuf< OUTPUT >:



Collaboration diagram for Rcpp::Rstreambuf< OUTPUT >:



## Public Member Functions

- [Rstreambuf](#) ()

## Protected Member Functions

- virtual std::streamsize [xspn](#) (const char \*s, std::streamsize n)
- virtual int [overflow](#) (int c=traits\_type::eof())
- virtual int [sync](#) ()
- std::streamsize [xspn](#) (const char \*s, std::streamsize num)
- std::streamsize [xspn](#) (const char \*s, std::streamsize num)
- int [overflow](#) (int c)
- int [overflow](#) (int c)
- int [sync](#) ()
- int [sync](#) ()

### 6.696.1 Detailed Description

```
template<bool OUTPUT>
class Rcpp::Rstreambuf< OUTPUT >
```

Definition at line 32 of file Rstreambuf.h.

### 6.696.2 Constructor & Destructor Documentation

#### 6.696.2.1 Rstreambuf()

```
template<bool OUTPUT>
Rcpp::Rstreambuf< OUTPUT >::Rstreambuf ( ) [inline]
```

Definition at line 34 of file Rstreambuf.h.

### 6.696.3 Member Function Documentation

#### 6.696.3.1 overflow() [1/3]

```
int Rcpp::Rstreambuf< true >::overflow (
    int c ) [inline], [protected]
```

Definition at line 61 of file Rstreambuf.h.

**6.696.3.2 overflow()** [2/3]

```
int Rcpp::Rstreambuf< false >::overflow (
    int c ) [inline], [protected]
```

Definition at line 68 of file Rstreambuf.h.

**6.696.3.3 overflow()** [3/3]

```
template<bool OUTPUT>
virtual int Rcpp::Rstreambuf< OUTPUT >::overflow (
    int c = traits_type::eof() ) [protected], [virtual]
```

**6.696.3.4 sync()** [1/3]

```
template<bool OUTPUT>
virtual int Rcpp::Rstreambuf< OUTPUT >::sync ( ) [protected], [virtual]
```

**6.696.3.5 sync()** [2/3]

```
int Rcpp::Rstreambuf< true >::sync ( ) [inline], [protected]
```

Definition at line 76 of file Rstreambuf.h.

**6.696.3.6 sync()** [3/3]

```
int Rcpp::Rstreambuf< false >::sync ( ) [inline], [protected]
```

Definition at line 80 of file Rstreambuf.h.

**6.696.3.7 xsputn()** [1/3]

```
template<bool OUTPUT>
virtual std::streamsize Rcpp::Rstreambuf< OUTPUT >::xsputn (
    const char * s,
    std::streamsize n ) [protected], [virtual]
```

**6.696.3.8 xspn()** [2/3]

```
std::streamsize Rcpp::Rstreambuf< true >::xspn (
    const char * s,
    std::streamsize num ) [inline], [protected]
```

Definition at line 52 of file Rstreambuf.h.

**6.696.3.9 xspn()** [3/3]

```
std::streamsize Rcpp::Rstreambuf< false >::xspn (
    const char * s,
    std::streamsize num ) [inline], [protected]
```

Definition at line 56 of file Rstreambuf.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/iostream/Rstreambuf.h

**6.697 Rcpp::algorithm::helpers::rtype< T > Struct Template Reference**

```
#include <algorithm.h>
```

**Public Types**

- typedef [rtype\\_helper](#)< typename [ctype](#)< T >::type >::type [type](#)
- typedef [rtype\\_helper](#)< typename [ctype](#)< T >::type > [helper\\_type](#)

**Static Public Member Functions**

- static T [NA](#) ()
- static [RCPP\\_CONSTEXPR\\_FUNC](#) T [ZERO](#) ()
- static [RCPP\\_CONSTEXPR\\_FUNC](#) T [ONE](#) ()

**Static Public Attributes**

- static [RCPP\\_CONSTEXPR\\_VAR](#) int [RTYPE](#) = [helper\\_type](#)::RTYPE

### 6.697.1 Detailed Description

```
template<typename T>
struct Rcpp::algorithm::helpers::rtype< T >
```

Definition at line 186 of file algorithm.h.

### 6.697.2 Member Typedef Documentation

#### 6.697.2.1 helper\_type

```
template<typename T >
typedef rtype_helper< typename ctype< T >::type > Rcpp::algorithm::helpers::rtype< T >::helper_type
```

Definition at line 188 of file algorithm.h.

#### 6.697.2.2 type

```
template<typename T >
typedef rtype_helper< typename ctype< T >::type >::type Rcpp::algorithm::helpers::rtype< T >↵
::type
```

Definition at line 187 of file algorithm.h.

### 6.697.3 Member Function Documentation

#### 6.697.3.1 NA()

```
template<typename T >
static T Rcpp::algorithm::helpers::rtype< T >::NA ( ) [inline], [static]
```

Definition at line 190 of file algorithm.h.

References Rcpp::NA.

Referenced by Rcpp::algorithm::mean(), Rcpp::algorithm::helpers::log::operator>(), Rcpp::algorithm::helpers::exp↵
::operator>(), and Rcpp::algorithm::helpers::sqrt::operator().



### 6.697.3.2 ONE()

```
template<typename T >
static RCPP_CONSTEXPR_FUNC T Rcpp::algorithm::helpers::rtype< T >::ONE ( ) [inline], [static]
```

Definition at line 192 of file algorithm.h.

### 6.697.3.3 ZERO()

```
template<typename T >
static RCPP_CONSTEXPR_FUNC T Rcpp::algorithm::helpers::rtype< T >::ZERO ( ) [inline], [static]
```

Definition at line 191 of file algorithm.h.

## 6.697.4 Member Data Documentation

### 6.697.4.1 RTYPE

```
template<typename T >
RCPP_CONSTEXPR_VAR int Rcpp::algorithm::helpers::rtype< T >::RTYPE = helper_type::RTYPE [static]
```

Definition at line 189 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.698 Rcpp::algorithm::helpers::rtype\_helper< T > Struct Template Reference

```
#include <algorithm.h>
```

### 6.698.1 Detailed Description

```
template<typename T>
struct Rcpp::algorithm::helpers::rtype_helper< T >
```

Definition at line 164 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.699 Rcpp::algorithm::helpers::rtype\_helper< double > Struct Reference

```
#include <algorithm.h>
```

### Public Types

- typedef double [type](#)

### Static Public Member Functions

- static double [NA](#) ()
- static [RCPP\\_CONSTEXPR\\_FUNC](#) double [ZERO](#) ()
- static [RCPP\\_CONSTEXPR\\_FUNC](#) double [ONE](#) ()

### Static Public Attributes

- static [RCPP\\_CONSTEXPR\\_VAR](#) int [RTYPE](#) = REALSXP

### 6.699.1 Detailed Description

Definition at line 168 of file `algorithm.h`.

### 6.699.2 Member Typedef Documentation

#### 6.699.2.1 type

```
typedef double Rcpp::algorithm::helpers::rtype\_helper< double >::type
```

Definition at line 169 of file `algorithm.h`.

### 6.699.3 Member Function Documentation

#### 6.699.3.1 NA()

```
static double Rcpp::algorithm::helpers::rtype\_helper< double >::NA ( ) [inline], [static]
```

Definition at line 171 of file `algorithm.h`.

### 6.699.3.2 ONE()

```
static RCPP_CONSTEXPR_FUNC double Rcpp::algorithm::helpers::rtype_helper< double >::ONE ( ) [inline],  
[static]
```

Definition at line 173 of file algorithm.h.

### 6.699.3.3 ZERO()

```
static RCPP_CONSTEXPR_FUNC double Rcpp::algorithm::helpers::rtype_helper< double >::ZERO ( )  
[inline], [static]
```

Definition at line 172 of file algorithm.h.

## 6.699.4 Member Data Documentation

### 6.699.4.1 RTYPE

```
RCPP_CONSTEXPR_VAR int Rcpp::algorithm::helpers::rtype_helper< double >::RTYPE = REALSXP [static]
```

Definition at line 170 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.700 Rcpp::algorithm::helpers::rtype\_helper< int > Struct Reference

```
#include <algorithm.h>
```

### Public Types

- typedef int [type](#)

### Static Public Member Functions

- static int [NA](#) ()
- static [RCPP\\_CONSTEXPR\\_FUNC](#) int [ZERO](#) ()
- static [RCPP\\_CONSTEXPR\\_FUNC](#) int [ONE](#) ()

## Static Public Attributes

- static `RCPP_CONSTEXPR_VAR` int `RTYPE` = INTSXP

### 6.700.1 Detailed Description

Definition at line 177 of file `algorithm.h`.

### 6.700.2 Member Typedef Documentation

#### 6.700.2.1 type

```
typedef int Rcpp::algorithm::helpers::rtype_helper< int >::type
```

Definition at line 178 of file `algorithm.h`.

### 6.700.3 Member Function Documentation

#### 6.700.3.1 NA()

```
static int Rcpp::algorithm::helpers::rtype_helper< int >::NA ( ) [inline], [static]
```

Definition at line 180 of file `algorithm.h`.

#### 6.700.3.2 ONE()

```
static RCPP_CONSTEXPR_FUNC int Rcpp::algorithm::helpers::rtype_helper< int >::ONE ( ) [inline],  
[static]
```

Definition at line 182 of file `algorithm.h`.

### 6.700.3.3 ZERO()

```
static RCPP_CONSTEXPR_FUNC int Rcpp::algorithm::helpers::rtype_helper< int >::ZERO ( ) [inline],  
[static]
```

Definition at line 181 of file algorithm.h.

## 6.700.4 Member Data Documentation

### 6.700.4.1 RTYPE

```
RCPP_CONSTEXPR_VAR int Rcpp::algorithm::helpers::rtype_helper< int >::RTYPE = INTSXP [static]
```

Definition at line 179 of file algorithm.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/algorithm.h](#)

## 6.701 Rcpp::rule Struct Reference

### Public Attributes

- [int r\\_type](#)
- [int r\\_day](#)
- [int r\\_week](#)
- [int r\\_mon](#)
- [int\\_fast32\\_t r\\_time](#)

### 6.701.1 Detailed Description

Definition at line 408 of file date.cpp.

### 6.701.2 Member Data Documentation

### 6.701.2.1 r\_day

```
int Rcpp::rule::r_day
```

Definition at line 410 of file date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

### 6.701.2.2 r\_mon

```
int Rcpp::rule::r_mon
```

Definition at line 412 of file date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

### 6.701.2.3 r\_time

```
int_fast32_t Rcpp::rule::r_time
```

Definition at line 413 of file date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

### 6.701.2.4 r\_type

```
int Rcpp::rule::r_type
```

Definition at line 409 of file date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

### 6.701.2.5 r\_week

```
int Rcpp::rule::r_week
```

Definition at line 411 of file date.cpp.

Referenced by Rcpp::getrule(), and Rcpp::transtime().

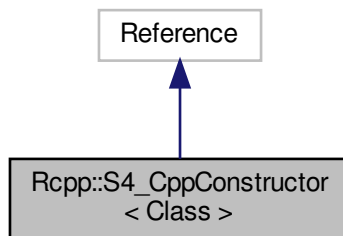
The documentation for this struct was generated from the following file:

- [src/date.cpp](#)

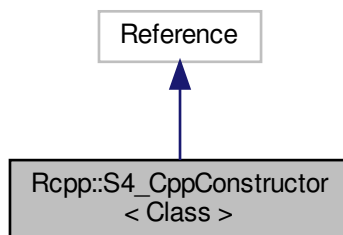
## 6.702 Rcpp::S4\_CppConstructor< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::S4\_CppConstructor< Class >:



Collaboration diagram for Rcpp::S4\_CppConstructor< Class >:



### Public Types

- typedef [XPtr< class\\_Base > XP\\_Class](#)
- typedef Reference::Storage [Storage](#)

### Public Member Functions

- [S4\\_CppConstructor](#) ([SignedConstructor](#)< Class > \*m, const [XP\\_Class](#) &class\_xp, const std::string &class\_name, std::string &buffer)

## Private Types

- typedef [Reference Base](#)

### 6.702.1 Detailed Description

```
template<typename Class >  
class Rcpp::S4_CppConstructor< Class >
```

Definition at line 203 of file Module.h.

### 6.702.2 Member Typedef Documentation

#### 6.702.2.1 Base

```
template<typename Class >  
typedef Reference Rcpp::S4\_CppConstructor< Class >::Base [private]
```

Definition at line 204 of file Module.h.

#### 6.702.2.2 Storage

```
template<typename Class >  
typedef Reference::Storage Rcpp::S4\_CppConstructor< Class >::Storage
```

Definition at line 207 of file Module.h.

#### 6.702.2.3 XP\_Class

```
template<typename Class >  
typedef XPTr<class\_Base> Rcpp::S4\_CppConstructor< Class >::XP\_Class
```

Definition at line 206 of file Module.h.

### 6.702.3 Constructor & Destructor Documentation



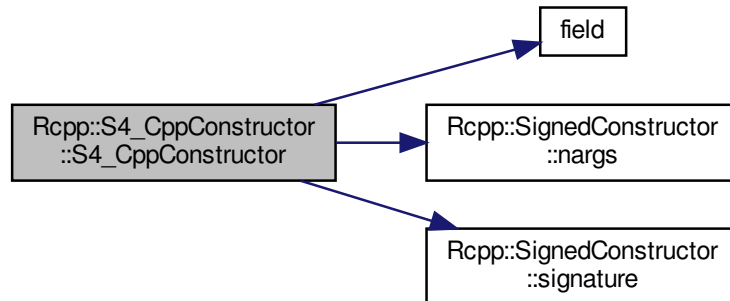
## 6.702.3.1 S4\_CppConstructor()

```
template<typename Class >
Rcpp::S4_CppConstructor< Class >::S4_CppConstructor (
    SignedConstructor< Class > * m,
    const XP_Class & class_xp,
    const std::string & class_name,
    std::string & buffer ) [inline]
```

Definition at line 209 of file Module.h.

References Rcpp::SignedConstructor< Class >::docstring, field(), Rcpp::SignedConstructor< Class >::nargs(), RCPP\_DEBUG, and Rcpp::SignedConstructor< Class >::signature().

Here is the call graph for this function:



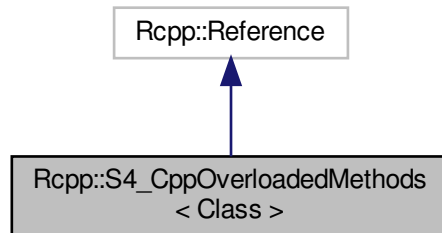
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Module.h](#)

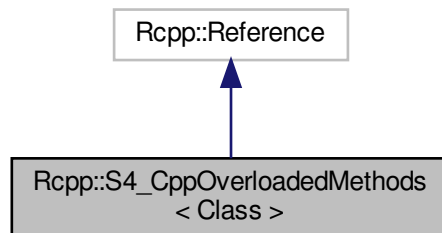
## 6.703 Rcpp::S4\_CppOverloadedMethods< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for `Rcpp::S4_CppOverloadedMethods< Class >`:



Collaboration diagram for `Rcpp::S4_CppOverloadedMethods< Class >`:



## Public Types

- typedef [Rcpp::XPtr< class\\_Base > XP\\_Class](#)
- typedef [SignedMethod< Class > signed\\_method\\_class](#)
- typedef `std::vector< signed\_method\_class \* >` [vec\\_signed\\_method](#)

## Public Member Functions

- [S4\\_CppOverloadedMethods](#) ([vec\\_signed\\_method](#) \*m, const [XP\\_Class](#) &class\_xp, const char \*name, std::string &buffer)

## Private Types

- typedef [Rcpp::Reference](#) Base

### 6.703.1 Detailed Description

```
template<typename Class >  
class Rcpp::S4_CppOverloadedMethods< Class >
```

Definition at line 224 of file Module.h.

### 6.703.2 Member Typedef Documentation

#### 6.703.2.1 Base

```
template<typename Class >  
typedef Rcpp::Reference Rcpp::S4_CppOverloadedMethods< Class >::Base [private]
```

Definition at line 225 of file Module.h.

#### 6.703.2.2 signed\_method\_class

```
template<typename Class >  
typedef SignedMethod<Class> Rcpp::S4_CppOverloadedMethods< Class >::signed_method_class
```

Definition at line 228 of file Module.h.

#### 6.703.2.3 vec\_signed\_method

```
template<typename Class >  
typedef std::vector<signed_method_class*> Rcpp::S4_CppOverloadedMethods< Class >::vec_signed_method
```

Definition at line 229 of file Module.h.

#### 6.703.2.4 XP\_Class

```
template<typename Class >  
typedef Rcpp::XPtr<class_Base> Rcpp::S4_CppOverloadedMethods< Class >::XP_Class
```

Definition at line 227 of file Module.h.

### 6.703.3 Constructor & Destructor Documentation

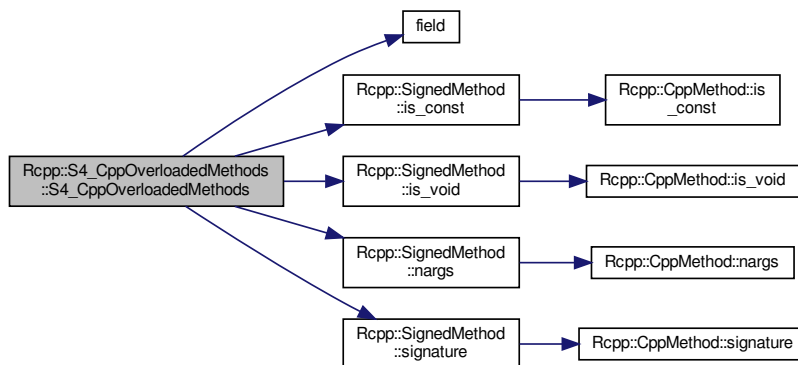
#### 6.703.3.1 S4\_CppOverloadedMethods()

```
template<typename Class >
Rcpp::S4_CppOverloadedMethods< Class >::S4_CppOverloadedMethods (
    vec_signed_method * m,
    const XP_Class & class_xp,
    const char * name,
    std::string & buffer ) [inline]
```

Definition at line 231 of file Module.h.

References `Rcpp::SignedMethod< Class >::docstring`, `field()`, `Rcpp::SignedMethod< Class >::is_const()`, `Rcpp::SignedMethod< Class >::is_void()`, `Rcpp::SignedMethod< Class >::nargs()`, and `Rcpp::SignedMethod< Class >::signature()`.

Here is the call graph for this function:



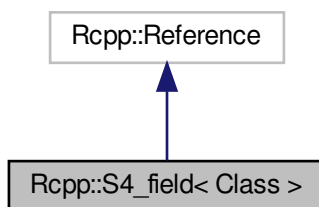
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Module.h](#)

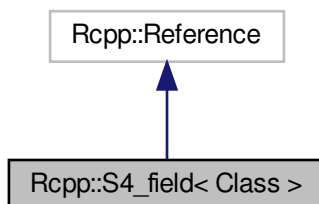
## 6.704 Rcpp::S4\_field< Class > Class Template Reference

```
#include <Module.h>
```

Inheritance diagram for Rcpp::S4\_field< Class >:



Collaboration diagram for Rcpp::S4\_field< Class >:



### Public Types

- typedef [XPtr< class\\_Base > XP\\_Class](#)

### Public Member Functions

- [S4\\_field](#) ([CppClassProperty< Class > \\*p](#), const [XP\\_Class](#) &class\_xp)

### Private Types

- typedef [Rcpp::Reference Base](#)

### 6.704.1 Detailed Description

```
template<typename Class>  
class Rcpp::S4_field< Class >
```

Definition at line 321 of file Module.h.

### 6.704.2 Member Typedef Documentation

#### 6.704.2.1 Base

```
template<typename Class >  
typedef Rcpp::Reference Rcpp::S4_field< Class >::Base [private]
```

Definition at line 322 of file Module.h.

#### 6.704.2.2 XP\_Class

```
template<typename Class >  
typedef XPtr<class_Base> Rcpp::S4_field< Class >::XP_Class
```

Definition at line 324 of file Module.h.

### 6.704.3 Constructor & Destructor Documentation

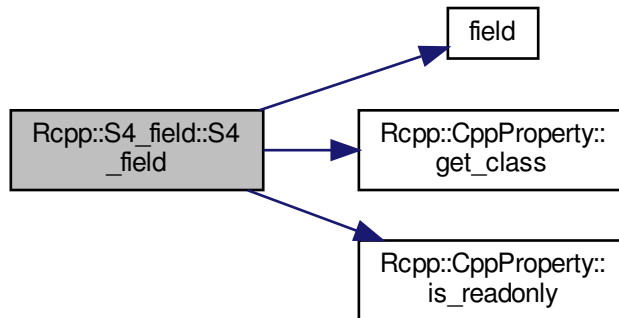
## 6.704.3.1 S4\_field()

```
template<typename Class >
Rcpp::S4_field< Class >::S4_field (
    CppProperty< Class > * p,
    const XP_Class & class_xp ) [inline]
```

Definition at line 325 of file Module.h.

References Rcpp::CppProperty< Class >::docstring, field(), Rcpp::CppProperty< Class >::get\_class(), Rcpp::CppProperty< Class >::is\_readonly(), and RCPP\_DEBUG.

Here is the call graph for this function:



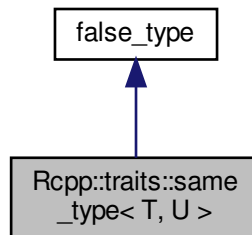
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Module.h](#)

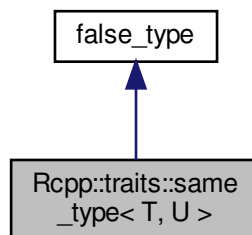
## 6.705 Rcpp::traits::same\_type&lt; T, U &gt; Struct Template Reference

```
#include <same_type.h>
```

Inheritance diagram for Rcpp::traits::same\_type< T, U >:



Collaboration diagram for Rcpp::traits::same\_type< T, U >:



## Additional Inherited Members

### 6.705.1 Detailed Description

```
template<typename T, typename U>  
struct Rcpp::traits::same_type< T, U >
```

Definition at line 29 of file same\_type.h.

The documentation for this struct was generated from the following file:

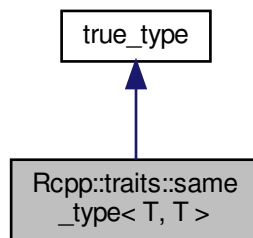
- [inst/include/Rcpp/traits/same\\_type.h](#)



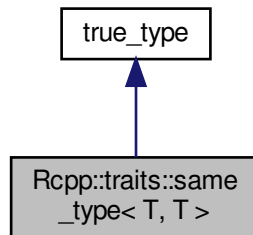
## 6.706 Rcpp::traits::same\_type< T, T > Struct Template Reference

```
#include <same_type.h>
```

Inheritance diagram for Rcpp::traits::same\_type< T, T >:



Collaboration diagram for Rcpp::traits::same\_type< T, T >:



### Additional Inherited Members

#### 6.706.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::same_type< T, T >
```

Definition at line 30 of file same\_type.h.

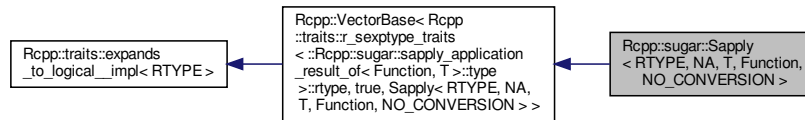
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[same\\_type.h](#)

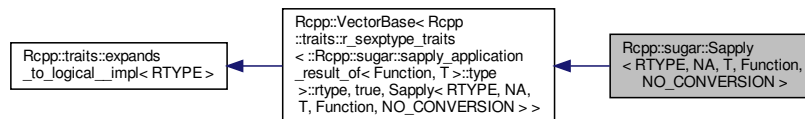
## 6.707 Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO\_CONVERSION > Class Template Reference

```
#include <sapply.h>
```

Inheritance diagram for Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO\_CONVERSION >:



Collaboration diagram for Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO\_CONVERSION >:



### Public Types

- typedef `::Rcpp::sugar::sapply_application_result_of< Function, T >::type` `result_type`
- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC`
- typedef `Rcpp::traits::r_vector_element_converter< RESULT_R_TYPE >::type` `converter_type`
- typedef `Rcpp::traits::storage_type< RESULT_R_TYPE >::type` `STORAGE`
- typedef `Rcpp::traits::Extractor< RTYPE, NA, T >::type` `EXT`

### Public Member Functions

- `Sapply` (const `VEC` &vec\_, `Function` fun\_)
- `STORAGE` `operator[]` (`R_xlen_t` i) const
- `R_xlen_t` `size` () const

### Static Public Attributes

- static const int `RESULT_R_TYPE`

## Private Attributes

- const [EXT](#) & [vec](#)
- [Function](#) [fun](#)

### 6.707.1 Detailed Description

```
template<int RTYPE, bool NA, typename T, typename Function, bool NO_CONVERSION>
class Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >
```

Definition at line 46 of file `sapply.h`.

### 6.707.2 Member Typedef Documentation

#### 6.707.2.1 converter\_type

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
typedef Rcpp::traits::r_vector_element_converter<RESULT_R_TYPE>::type Rcpp::sugar::Sapply< RTYPE,
NA, T, Function, NO_CONVERSION >::converter_type
```

Definition at line 59 of file `sapply.h`.

#### 6.707.2.2 EXT

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Sapply< RTYPE, NA, T, Function,
NO_CONVERSION >::EXT
```

Definition at line 62 of file `sapply.h`.

#### 6.707.2.3 result\_type

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
typedef ::Rcpp::sugar::sapply_application_result_of<Function, T>::type Rcpp::sugar::Sapply< RTYPE,
NA, T, Function, NO_CONVERSION >::result_type
```

Definition at line 54 of file `sapply.h`.

### 6.707.2.4 STORAGE

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
typedef Rcpp::traits::storage_type<RESULT_R_TYPE>::type Rcpp::sugar::Sapply< RTYPE, NA, T, Function,
NO_CONVERSION >::STORAGE
```

Definition at line 60 of file sapply.h.

### 6.707.2.5 VEC

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION
>::VEC
```

Definition at line 58 of file sapply.h.

## 6.707.3 Constructor & Destructor Documentation

### 6.707.3.1 Sapply()

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::Sapply (
    const VEC & vec_,
    Function fun_ ) [inline]
```

Definition at line 64 of file sapply.h.

References DEMANGLE, and RCPP\_DEBUG\_1.

## 6.707.4 Member Function Documentation

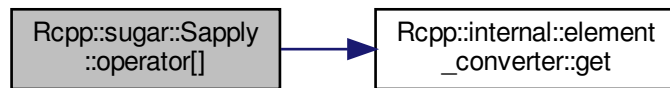
### 6.707.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
STORAGE Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 69 of file sapply.h.

References Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO\_CONVERSION >::fun, Rcpp::internal::element\_↔\_converter< RTYPE >::get(), and Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO\_CONVERSION >::vec.

Here is the call graph for this function:



### 6.707.4.2 size()

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
R_xlen_t Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::size ( ) const [inline]
```

Definition at line 73 of file sapply.h.

References Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO\_CONVERSION >::vec.

## 6.707.5 Member Data Documentation

### 6.707.5.1 fun

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
Function Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::fun [private]
```

Definition at line 77 of file sapply.h.

Referenced by Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO\_CONVERSION >::operator[](), and Rcpp::sugar↔::Sapply< RTYPE, NA, T, Function, true >::operator[]().

### 6.707.5.2 RESULT\_R\_TYPE

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
const int Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::RESULT_R_TYPE [static]
```

#### Initial value:

```
=
    Rcpp::traits::r_sexptype_traits<result_type>::rtype
```

Definition at line 55 of file sapply.h.

### 6.707.5.3 vec

```
template<int RTYPE, bool NA, typename T , typename Function , bool NO_CONVERSION>
const EXT& Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::vec [private]
```

Definition at line 76 of file sapply.h.

Referenced by `Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::operator[]()`, `Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::operator[]()`, `Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO_CONVERSION >::size()`, and `Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::size()`.

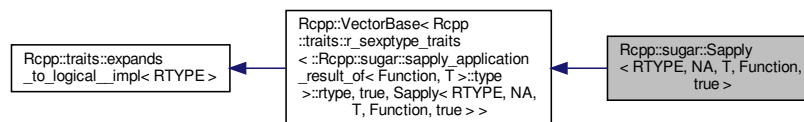
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/sapply.h`

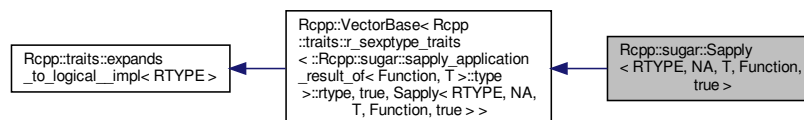
## 6.708 Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true > Class Template Reference

```
#include <sapply.h>
```

Inheritance diagram for `Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >`:



Collaboration diagram for `Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >`:



## Public Types

- typedef `Rcpp::sugar::sapply_application_result_of< Function, T >::type` `result_type`
- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC`
- typedef `Rcpp::traits::storage_type< RESULT_R_TYPE >::type` `STORAGE`
- typedef `Rcpp::traits::Extractor< RTYPE, NA, T >::type` `EXT`

## Public Member Functions

- `Sapply` (const `VEC` &vec\_, `Function` fun\_)
- `STORAGE operator[]` (`R_xlen_t` i) const
- `R_xlen_t size` () const

## Static Public Attributes

- static const int `RESULT_R_TYPE`

## Private Attributes

- const `EXT` & `vec`
- `Function` fun

### 6.708.1 Detailed Description

```
template<int RTYPE, bool NA, typename T, typename Function>
class Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >
```

Definition at line 83 of file `sapply.h`.

### 6.708.2 Member Typedef Documentation

#### 6.708.2.1 EXT

```
template<int RTYPE, bool NA, typename T , typename Function >
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Sapply< RTYPE, NA, T, Function,
true >::EXT
```

Definition at line 98 of file `sapply.h`.

### 6.708.2.2 result\_type

```
template<int RTYPE, bool NA, typename T , typename Function >
typedef ::Rcpp::sugar::sapply_application_result_of<Function, T>::type Rcpp::sugar::Sapply< RTYPE,
NA, T, Function, true >::result_type
```

Definition at line 91 of file sapply.h.

### 6.708.2.3 STORAGE

```
template<int RTYPE, bool NA, typename T , typename Function >
typedef Rcpp::traits::storage_type<RESULT_R_TYPE>::type Rcpp::sugar::Sapply< RTYPE, NA, T, Function,
true >::STORAGE
```

Definition at line 96 of file sapply.h.

### 6.708.2.4 VEC

```
template<int RTYPE, bool NA, typename T , typename Function >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::VEC
```

Definition at line 95 of file sapply.h.

## 6.708.3 Constructor & Destructor Documentation

### 6.708.3.1 Sapply()

```
template<int RTYPE, bool NA, typename T , typename Function >
Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::Sapply (
    const VEC & vec_,
    Function fun_ ) [inline]
```

Definition at line 100 of file sapply.h.

References DEMANGLE, and RCPP\_DEBUG\_1.

### 6.708.4 Member Function Documentation



### 6.708.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T , typename Function >
STORAGE Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 104 of file sapply.h.

References Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO\_CONVERSION >::fun, and Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO\_CONVERSION >::vec.

### 6.708.4.2 size()

```
template<int RTYPE, bool NA, typename T , typename Function >
R_xlen_t Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::size ( ) const [inline]
```

Definition at line 107 of file sapply.h.

References Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO\_CONVERSION >::vec.

## 6.708.5 Member Data Documentation

### 6.708.5.1 fun

```
template<int RTYPE, bool NA, typename T , typename Function >
Function Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::fun [private]
```

Definition at line 111 of file sapply.h.

### 6.708.5.2 RESULT\_R\_TYPE

```
template<int RTYPE, bool NA, typename T , typename Function >
const int Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::RESULT_R_TYPE [static]
```

#### Initial value:

```
=
    Rcpp::traits::r_sexptype_traits<result_type>::rtype
```

Definition at line 92 of file sapply.h.

### 6.708.5.3 vec

```
template<int RTYPE, bool NA, typename T , typename Function >  
const EXT& Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >::vec [private]
```

Definition at line 110 of file sapply.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/sapply.h](#)

## 6.709 Rcpp::sugar::sapply\_application\_result\_of< Function, SugarExpression > Struct Template Reference

```
#include <sapply.h>
```

### Public Types

- typedef [::Rcpp::traits::result\\_of< Function >::type](#) type

### 6.709.1 Detailed Description

```
template<typename Function, typename SugarExpression>  
struct Rcpp::sugar::sapply_application_result_of< Function, SugarExpression >
```

Definition at line 33 of file sapply.h.

### 6.709.2 Member Typedef Documentation

#### 6.709.2.1 type

```
template<typename Function , typename SugarExpression >  
typedef ::Rcpp::traits::result\_of<Function>::type Rcpp::sugar::sapply_application_result_of<  
Function, SugarExpression >::type
```

Definition at line 38 of file sapply.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/functions/sapply.h](#)

## 6.710 Rcpp::sugar::cbind\_impl::scalar< RTYPE > Struct Template Reference

```
#include <cbind.h>
```

### Public Types

- typedef [cbind\\_storage\\_type](#)< RTYPE >::type type

### 6.710.1 Detailed Description

```
template<int RTYPE>  
struct Rcpp::sugar::cbind_impl::scalar< RTYPE >
```

Definition at line 129 of file cbind.h.

### 6.710.2 Member Typedef Documentation

#### 6.710.2.1 type

```
template<int RTYPE>  
typedef cbind\_storage\_type<RTYPE>::type Rcpp::sugar::cbind_impl::scalar< RTYPE >::type
```

Definition at line 130 of file cbind.h.

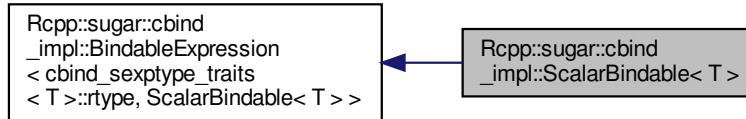
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/sugar/functions/[cbind.h](#)

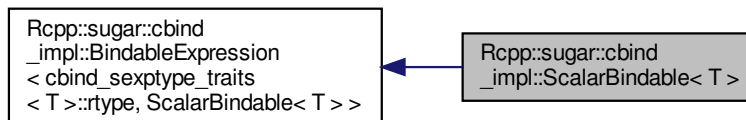
## 6.711 Rcpp::sugar::cbind\_impl::ScalarBindable< T > Class Template Reference

```
#include <cbind.h>
```

Inheritance diagram for Rcpp::sugar::cbind\_impl::ScalarBindable< T >:



Collaboration diagram for Rcpp::sugar::cbind\_impl::ScalarBindable< T >:



### Public Types

- enum { [RTYPE](#) = cbind\_sexptype\_traits<T>::rtype }
- typedef T [stored\\_type](#)

### Public Member Functions

- [ScalarBindable](#) (const T &t\_)
- [R\\_xlen\\_t size](#) () const
- [R\\_xlen\\_t nrow](#) () const
- [R\\_xlen\\_t ncol](#) () const
- [stored\\_type operator\[\]](#) (R\_xlen\_t i) const
- [stored\\_type operator\(\)](#) (R\_xlen\_t i, R\_xlen\_t j) const

### Private Attributes

- T t

## 6.711.1 Detailed Description

```
template<typename T>
class Rcpp::sugar::cbind_impl::ScalarBindable< T >
```

Definition at line 135 of file cbind.h.

## 6.711.2 Member Typedef Documentation

### 6.711.2.1 stored\_type

```
template<typename T >
typedef T Rcpp::sugar::cbind_impl::ScalarBindable< T >::stored_type
```

Definition at line 139 of file cbind.h.

## 6.711.3 Member Enumeration Documentation

### 6.711.3.1 anonymous enum

```
template<typename T >
anonymous enum
```

#### Enumerator

RTYPE
-------

Definition at line 140 of file cbind.h.

## 6.711.4 Constructor & Destructor Documentation

### 6.711.4.1 ScalarBindable()

```
template<typename T >
Rcpp::sugar::cbind_impl::ScalarBindable< T >::ScalarBindable (
    const T & t_ ) [inline]
```

Definition at line 146 of file cbind.h.

## 6.711.5 Member Function Documentation

### 6.711.5.1 ncol()

```
template<typename T >  
R_xlen_t Rcpp::sugar::cbind_impl::ScalarBindable< T >::ncol ( ) const [inline]
```

Definition at line 152 of file cbind.h.

### 6.711.5.2 nrow()

```
template<typename T >  
R_xlen_t Rcpp::sugar::cbind_impl::ScalarBindable< T >::nrow ( ) const [inline]
```

Definition at line 150 of file cbind.h.

### 6.711.5.3 operator()()

```
template<typename T >  
stored_type Rcpp::sugar::cbind_impl::ScalarBindable< T >::operator() (   
    R_xlen_t i,  
    R_xlen_t j ) const [inline]
```

Definition at line 158 of file cbind.h.

References [Rcpp::sugar::cbind\\_impl::ScalarBindable< T >::t](#).

### 6.711.5.4 operator[]()

```
template<typename T >  
stored_type Rcpp::sugar::cbind_impl::ScalarBindable< T >::operator[] (   
    R_xlen_t i ) const [inline]
```

Definition at line 154 of file cbind.h.

References [Rcpp::sugar::cbind\\_impl::ScalarBindable< T >::t](#).

## 6.711.5.5 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::cbind_impl::ScalarBindable< T >::size ( ) const [inline]
```

Definition at line 148 of file cbind.h.

## 6.711.6 Member Data Documentation

## 6.711.6.1 t

```
template<typename T >
T Rcpp::sugar::cbind_impl::ScalarBindable< T >::t [private]
```

Definition at line 143 of file cbind.h.

Referenced by Rcpp::sugar::cbind\_impl::ScalarBindable< T >::operator>(), and Rcpp::sugar::cbind\_impl::ScalarBindable< T >::operator[]().

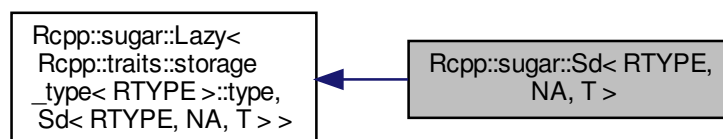
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/cbind.h

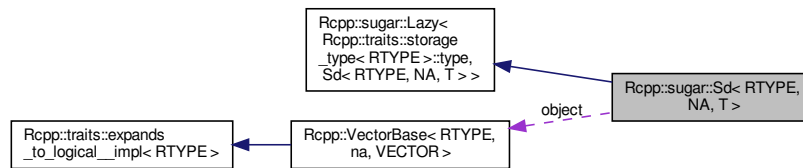
## 6.712 Rcpp::sugar::Sd&lt; RTYPE, NA, T &gt; Class Template Reference

```
#include <sd.h>
```

Inheritance diagram for Rcpp::sugar::Sd< RTYPE, NA, T >:



Collaboration diagram for `Rcpp::sugar::Sd< RTYPE, NA, T >`:



## Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`

## Public Member Functions

- `Sd` (const `VEC_TYPE` &object\_)
- `STORAGE` `get` () const

## Private Attributes

- const `VEC_TYPE` & `object`

### 6.712.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Sd< RTYPE, NA, T >
```

Definition at line 29 of file `sd.h`.

### 6.712.2 Member Typedef Documentation

#### 6.712.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Sd< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file `sd.h`.



### 6.712.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Sd< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file sd.h.

## 6.712.3 Constructor & Destructor Documentation

### 6.712.3.1 Sd()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Sd< RTYPE, NA, T >::Sd (
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 34 of file sd.h.

## 6.712.4 Member Function Documentation

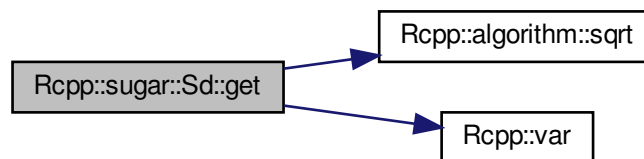
### 6.712.4.1 get()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Sd< RTYPE, NA, T >::get ( ) const [inline]
```

Definition at line 36 of file sd.h.

References Rcpp::algorithm::sqrt(), and Rcpp::var().

Here is the call graph for this function:



## 6.712.5 Member Data Documentation

### 6.712.5.1 object

```
template<int RTYPE, bool NA, typename T >
const VEC\_TYPE& Rcpp::sugar::Sd< RTYPE, NA, T >::object [private]
```

Definition at line 40 of file `sd.h`.

The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/sd.h`

## 6.713 [Rcpp::sugar::SelfHash](#)< RTYPE > Class Template Reference

```
#include <SelfHash.h>
```

### Public Types

- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Vector](#)< RTYPE > [VECTOR](#)

### Public Member Functions

- [SelfHash](#) (SEXP table)
- [IntegerVector fill\\_and\\_self\\_match](#) ()
- int [size](#) () const
- int [add\\_value\\_get\\_index](#) (int i)
- unsigned int [get\\_index](#) ([STORAGE](#) value) const
- unsigned int [get\\_addr](#) ([STORAGE](#) value) const
- unsigned int [get\\_addr](#) (int value) const
- unsigned int [get\\_addr](#) (double val) const
- unsigned int [get\\_addr](#) (SEXP value) const

### Public Attributes

- int [n](#)
- int [m](#)
- int [k](#)
- [STORAGE](#) \* [src](#)
- `std::vector< int >` [data](#)
- `std::vector< int >` [indices](#)
- int [size\\_](#)

## 6.713.1 Detailed Description

```
template<int RTYPE>
class Rcpp::sugar::SelfHash< RTYPE >
```

Definition at line 32 of file SelfHash.h.

## 6.713.2 Member Typedef Documentation

### 6.713.2.1 STORAGE

```
template<int RTYPE>
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::SelfHash< RTYPE >::STORAGE
```

Definition at line 34 of file SelfHash.h.

### 6.713.2.2 VECTOR

```
template<int RTYPE>
typedef Vector<RTYPE> Rcpp::sugar::SelfHash< RTYPE >::VECTOR
```

Definition at line 35 of file SelfHash.h.

## 6.713.3 Constructor & Destructor Documentation

### 6.713.3.1 SelfHash()

```
template<int RTYPE>
Rcpp::sugar::SelfHash< RTYPE >::SelfHash (
    SEXP table ) [inline]
```

Definition at line 37 of file SelfHash.h.

References Rcpp::sugar::SelfHash< RTYPE >::data, Rcpp::sugar::SelfHash< RTYPE >::indices, Rcpp::sugar::SelfHash< RTYPE >::k, Rcpp::sugar::SelfHash< RTYPE >::m, and Rcpp::sugar::SelfHash< RTYPE >::n.

## 6.713.4 Member Function Documentation

### 6.713.4.1 add\_value\_get\_index()

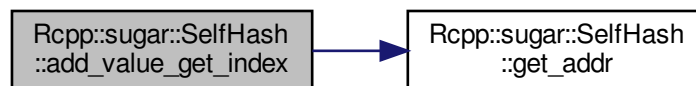
```
template<int RTYPE>
int Rcpp::sugar::SelfHash< RTYPE >::add_value_get_index (
    int i ) [inline]
```

Definition at line 63 of file SelfHash.h.

References `Rcpp::sugar::SelfHash< RTYPE >::data`, `Rcpp::sugar::SelfHash< RTYPE >::get_addr()`, `Rcpp::sugar::SelfHash< RTYPE >::indices`, `Rcpp::sugar::SelfHash< RTYPE >::m`, `Rcpp::sugar::SelfHash< RTYPE >::size_`, and `Rcpp::sugar::SelfHash< RTYPE >::src`.

Referenced by `Rcpp::sugar::SelfHash< RTYPE >::fill_and_self_match()`.

Here is the call graph for this function:



### 6.713.4.2 fill\_and\_self\_match()

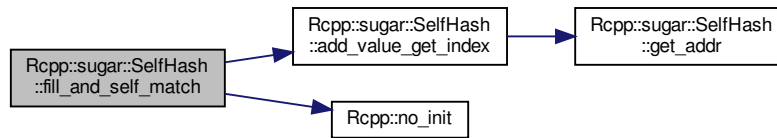
```
template<int RTYPE>
IntegerVector Rcpp::sugar::SelfHash< RTYPE >::fill_and_self_match ( ) [inline]
```

Definition at line 46 of file SelfHash.h.

References `Rcpp::sugar::SelfHash< RTYPE >::add_value_get_index()`, `Rcpp::sugar::SelfHash< RTYPE >::n`, and `Rcpp::no_init()`.

Referenced by `Rcpp::self_match()`.

Here is the call graph for this function:



#### 6.713.4.3 `get_addr()` [1/4]

```
unsigned int Rcpp::sugar::SelfHash< REALSXP >::get_addr (
    double val ) const [inline]
```

Definition at line 98 of file SelfHash.h.

References RCPP\_HASH.

#### 6.713.4.4 `get_addr()` [2/4]

```
unsigned int Rcpp::sugar::SelfHash< INTSXP >::get_addr (
    int value ) const [inline]
```

Definition at line 94 of file SelfHash.h.

References RCPP\_HASH.

#### 6.713.4.5 `get_addr()` [3/4]

```
unsigned int Rcpp::sugar::SelfHash< STRSXP >::get_addr (
    SEXP value ) const [inline]
```

Definition at line 115 of file SelfHash.h.

References RCPP\_HASH.

**6.713.4.6 get\_addr() [4/4]**

```
template<int RTYPE>
unsigned int Rcpp::sugar::SelfHash< RTYPE >::get_addr (
    STORAGE value ) const
```

Referenced by Rcpp::sugar::SelfHash< RTYPE >::add\_value\_get\_index(), and Rcpp::sugar::SelfHash< RTYPE >::get\_index().

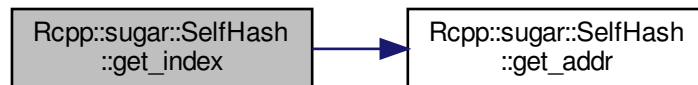
**6.713.4.7 get\_index()**

```
template<int RTYPE>
unsigned int Rcpp::sugar::SelfHash< RTYPE >::get_index (
    STORAGE value ) const [inline]
```

Definition at line 78 of file SelfHash.h.

References Rcpp::sugar::SelfHash< RTYPE >::data, Rcpp::sugar::SelfHash< RTYPE >::get\_addr(), Rcpp::sugar::SelfHash< RTYPE >::m, and Rcpp::sugar::SelfHash< RTYPE >::src.

Here is the call graph for this function:

**6.713.4.8 size()**

```
template<int RTYPE>
int Rcpp::sugar::SelfHash< RTYPE >::size ( ) const [inline]
```

Definition at line 53 of file SelfHash.h.

References Rcpp::sugar::SelfHash< RTYPE >::size\_.

**6.713.5 Member Data Documentation**

### 6.713.5.1 data

```
template<int RTYPE>
std::vector<int> Rcpp::sugar::SelfHash< RTYPE >::data
```

Definition at line 59 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::add\_value\_get\_index(), Rcpp::sugar::SelfHash< RTYPE >::get\_index(), and Rcpp::sugar::SelfHash< RTYPE >::SelfHash().

### 6.713.5.2 indices

```
template<int RTYPE>
std::vector<int> Rcpp::sugar::SelfHash< RTYPE >::indices
```

Definition at line 60 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::add\_value\_get\_index(), and Rcpp::sugar::SelfHash< RTYPE >::SelfHash().

### 6.713.5.3 k

```
template<int RTYPE>
int Rcpp::sugar::SelfHash< RTYPE >::k
```

Definition at line 57 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::SelfHash().

### 6.713.5.4 m

```
template<int RTYPE>
int Rcpp::sugar::SelfHash< RTYPE >::m
```

Definition at line 57 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::add\_value\_get\_index(), Rcpp::sugar::SelfHash< RTYPE >::get\_index(), and Rcpp::sugar::SelfHash< RTYPE >::SelfHash().

### 6.713.5.5 n

```
template<int RTYPE>
int Rcpp::sugar::SelfHash< RTYPE >::n
```

Definition at line 57 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::fill\_and\_self\_match(), and Rcpp::sugar::SelfHash< RTYPE >::SelfHash().

### 6.713.5.6 size\_

```
template<int RTYPE>
int Rcpp::sugar::SelfHash< RTYPE >::size_
```

Definition at line 61 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::add\_value\_get\_index(), and Rcpp::sugar::SelfHash< RTYPE >::size().

### 6.713.5.7 src

```
template<int RTYPE>
STORAGE* Rcpp::sugar::SelfHash< RTYPE >::src
```

Definition at line 58 of file SelfHash.h.

Referenced by Rcpp::sugar::SelfHash< RTYPE >::add\_value\_get\_index(), and Rcpp::sugar::SelfHash< RTYPE >::get\_index().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/hash/SelfHash.h](#)

## 6.714 Rcpp::sugar::SelfInserter< HASH, STORAGE > Class Template Reference

```
#include <self_match.h>
```



## Public Member Functions

- [SelfInserter](#) (HASH &hash\_)
- [R\\_xlen\\_t operator\(\)](#) (STORAGE value)

## Private Attributes

- HASH & [hash](#)
- [R\\_xlen\\_t index](#)

### 6.714.1 Detailed Description

```
template<typename HASH, typename STORAGE>
class Rcpp::sugar::SelfInserter< HASH, STORAGE >
```

Definition at line 29 of file self\_match.h.

### 6.714.2 Constructor & Destructor Documentation

#### 6.714.2.1 SelfInserter()

```
template<typename HASH , typename STORAGE >
Rcpp::sugar::SelfInserter< HASH, STORAGE >::SelfInserter (
    HASH & hash_ ) [inline]
```

Definition at line 31 of file self\_match.h.

### 6.714.3 Member Function Documentation

#### 6.714.3.1 operator>()

```
template<typename HASH , typename STORAGE >
R_xlen_t Rcpp::sugar::SelfInserter< HASH, STORAGE >::operator() (
    STORAGE value ) [inline]
```

Definition at line 33 of file self\_match.h.

References [Rcpp::sugar::SelfInserter< HASH, STORAGE >::hash](#), and [Rcpp::sugar::SelfInserter< HASH, STORAGE >::index](#).

## 6.714.4 Member Data Documentation

### 6.714.4.1 hash

```
template<typename HASH , typename STORAGE >
HASH& Rcpp::sugar::SelfInserter< HASH, STORAGE >::hash [private]
```

Definition at line 44 of file self\_match.h.

Referenced by Rcpp::sugar::SelfInserter< HASH, STORAGE >::operator().

### 6.714.4.2 index

```
template<typename HASH , typename STORAGE >
R_xlen_t Rcpp::sugar::SelfInserter< HASH, STORAGE >::index [private]
```

Definition at line 45 of file self\_match.h.

Referenced by Rcpp::sugar::SelfInserter< HASH, STORAGE >::operator().

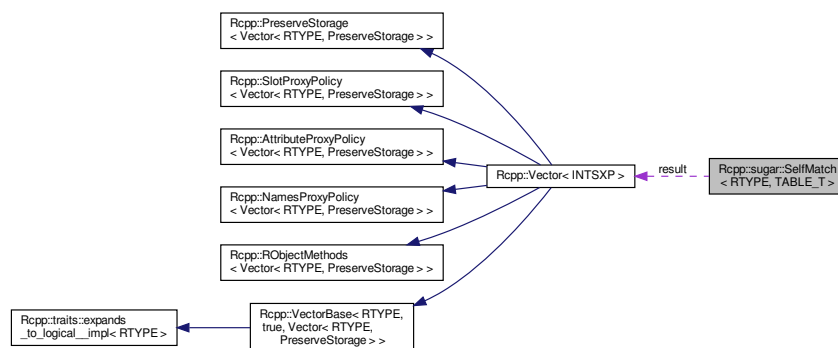
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/self\_match.h

## 6.715 Rcpp::sugar::SelfMatch< RTYPE, TABLE\_T > Class Template Reference

```
#include <self_match.h>
```

Collaboration diagram for Rcpp::sugar::SelfMatch< RTYPE, TABLE\_T >:



## Public Types

- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)

## Public Member Functions

- [SelfMatch](#) (const TABLE\_T &table)
- [operator IntegerVector](#) () const

## Private Types

- typedef [RCPP\\_UNORDERED\\_MAP](#)< [STORAGE](#), int > [HASH](#)
- typedef [SelfInserter](#)< [HASH](#), [STORAGE](#) > [Inserter](#)

## Private Attributes

- [HASH](#) hash
- [IntegerVector](#) result

### 6.715.1 Detailed Description

```
template<int RTYPE, typename TABLE_T>
class Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >
```

Definition at line 49 of file self\_match.h.

### 6.715.2 Member Typedef Documentation

#### 6.715.2.1 HASH

```
template<int RTYPE, typename TABLE_T >
typedef RCPP\_UNORDERED\_MAP<STORAGE, int> Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::HASH [private]
```

Definition at line 60 of file self\_match.h.

### 6.715.2.2 Inserter

```
template<int RTYPE, typename TABLE_T >
typedef SelfInserter<HASH, STORAGE> Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::Inserter [private]
```

Definition at line 61 of file self\_match.h.

### 6.715.2.3 STORAGE

```
template<int RTYPE, typename TABLE_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::STORAGE
```

Definition at line 51 of file self\_match.h.

## 6.715.3 Constructor & Destructor Documentation

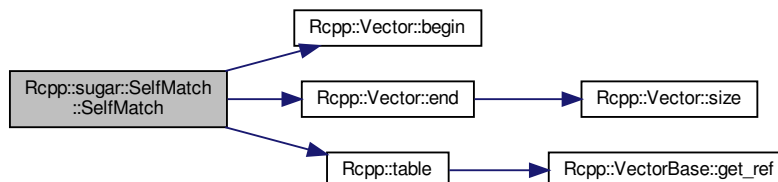
### 6.715.3.1 SelfMatch()

```
template<int RTYPE, typename TABLE_T >
Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::SelfMatch (
    const TABLE_T & table ) [inline]
```

Definition at line 53 of file self\_match.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::hash`, and `Rcpp::table()`.

Here is the call graph for this function:



## 6.715.4 Member Function Documentation

### 6.715.4.1 operator IntegerVector()

```
template<int RTYPE, typename TABLE_T >  
Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::operator IntegerVector ( ) const [inline]
```

Definition at line 57 of file self\_match.h.

References Rcpp::sugar::SelfMatch< RTYPE, TABLE\_T >::result.

## 6.715.5 Member Data Documentation

### 6.715.5.1 hash

```
template<int RTYPE, typename TABLE_T >  
HASH Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::hash [private]
```

Definition at line 62 of file self\_match.h.

Referenced by Rcpp::sugar::SelfMatch< RTYPE, TABLE\_T >::SelfMatch().

### 6.715.5.2 result

```
template<int RTYPE, typename TABLE_T >  
IntegerVector Rcpp::sugar::SelfMatch< RTYPE, TABLE_T >::result [private]
```

Definition at line 63 of file self\_match.h.

Referenced by Rcpp::sugar::SelfMatch< RTYPE, TABLE\_T >::operator IntegerVector().

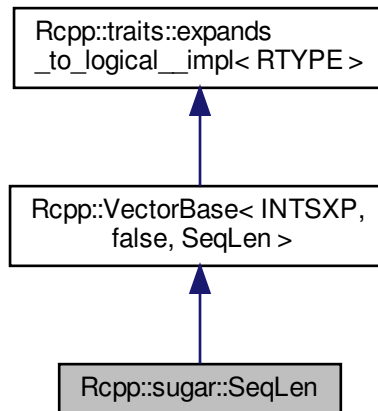
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/self\_match.h

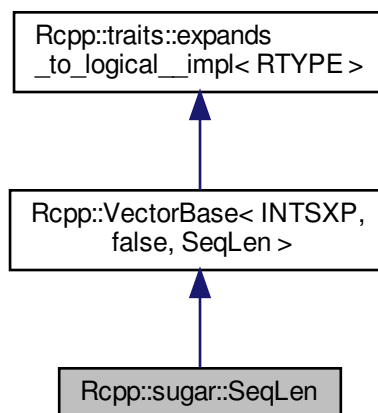
## 6.716 Rcpp::sugar::SeqLen Class Reference

```
#include <seq_along.h>
```

Inheritance diagram for Rcpp::sugar::SeqLen:



Collaboration diagram for Rcpp::sugar::SeqLen:



## Public Member Functions

- [SeqLen](#) (R\_xlen\_t len\_)
- R\_xlen\_t [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- R\_xlen\_t [len](#)

## Additional Inherited Members

### 6.716.1 Detailed Description

Definition at line 28 of file seq\_along.h.

### 6.716.2 Constructor & Destructor Documentation

#### 6.716.2.1 SeqLen()

```
Rcpp::sugar::SeqLen::SeqLen (  
    R_xlen_t len_ ) [inline]
```

Definition at line 30 of file seq\_along.h.

### 6.716.3 Member Function Documentation

#### 6.716.3.1 operator[]()

```
R_xlen_t Rcpp::sugar::SeqLen::operator[] (  
    R_xlen_t i ) const [inline]
```

Definition at line 32 of file seq\_along.h.

### 6.716.3.2 size()

```
R_xlen_t Rcpp::sugar::SeqLen::size ( ) const [inline]
```

Definition at line 35 of file seq\_along.h.

References len.

## 6.716.4 Member Data Documentation

### 6.716.4.1 len

```
R_xlen_t Rcpp::sugar::SeqLen::len [private]
```

Definition at line 38 of file seq\_along.h.

Referenced by size().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/seq\_along.h

## 6.717 Rcpp::sugar::SetDiff< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <setdiff.h>
```

### Public Types

- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)

### Public Member Functions

- [SetDiff](#) (const LHS\_T &lhs, const RHS\_T &rhs)
- [Vector](#)< RTYPE > [get](#) () const

### Private Types

- typedef [RCPP\\_UNORDERED\\_SET](#)< [STORAGE](#) > [SET](#)
- typedef [SET](#)::const\_iterator [ITERATOR](#)



## Private Attributes

- [SET lhs\\_set](#)
- [SET rhs\\_set](#)

### 6.717.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 43 of file setdiff.h.

### 6.717.2 Member Typedef Documentation

#### 6.717.2.1 ITERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SET::const_iterator Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::ITERATOR
[private]
```

Definition at line 64 of file setdiff.h.

#### 6.717.2.2 SET

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef RCPP_UNORDERED_SET<STORAGE> Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_↵
T >::SET [private]
```

Definition at line 63 of file setdiff.h.

#### 6.717.2.3 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_↵
_NA, RHS_T >::STORAGE
```

Definition at line 45 of file setdiff.h.

## 6.717.3 Constructor & Destructor Documentation

### 6.717.3.1 SetDiff()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::SetDiff (
    const LHS_T & lhs,
    const RHS_T & rhs ) [inline]
```

Definition at line 47 of file setdiff.h.

References `Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs_set`, and `Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs_set`.

## 6.717.4 Member Function Documentation

### 6.717.4.1 get()

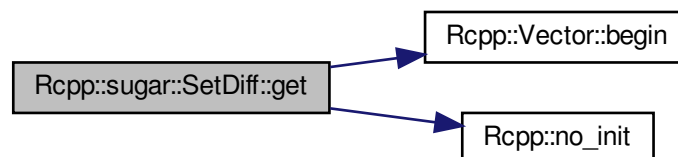
```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Vector<RTYPE> Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get ( ) const [inline]
```

Definition at line 55 of file setdiff.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs_set`, and `Rcpp::no_init()`.

Referenced by `Rcpp::setdiff()`.

Here is the call graph for this function:



## 6.717.5 Member Data Documentation

### 6.717.5.1 lhs\_set

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
SET Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs_set [private]
```

Definition at line 65 of file setdiff.h.

Referenced by Rcpp::sugar::SetDiff< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::get(), and Rcpp::sugar::SetDiff< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::SetDiff().

### 6.717.5.2 rhs\_set

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
SET Rcpp::sugar::SetDiff< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs_set [private]
```

Definition at line 66 of file setdiff.h.

Referenced by Rcpp::sugar::SetDiff< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::SetDiff().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[setdiff.h](#)

## 6.718 Rcpp::sugar::SetEqual< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <setdiff.h>
```

### Public Types

- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type STORAGE

### Public Member Functions

- [SetEqual](#) (const LHS\_T &lhs, const RHS\_T &rhs)
- bool [get](#) () const

## Private Types

- typedef [RCPP\\_UNORDERED\\_SET](#)< [STORAGE](#) > [SET](#)
- typedef [SET](#)::const\_iterator [ITERATOR](#)

## Private Attributes

- [SET](#) lhs\_set
- [SET](#) rhs\_set

### 6.718.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>  
class Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 71 of file setdiff.h.

### 6.718.2 Member Typedef Documentation

#### 6.718.2.1 ITERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef SET::const_iterator Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::ITERATOR  
[private]
```

Definition at line 93 of file setdiff.h.

#### 6.718.2.2 SET

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef RCPP\_UNORDERED\_SET<STORAGE> Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::SET [private]
```

Definition at line 92 of file setdiff.h.

### 6.718.2.3 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 73 of file setdiff.h.

## 6.718.3 Constructor & Destructor Documentation

### 6.718.3.1 SetEqual()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::SetEqual (
    const LHS_T & lhs,
    const RHS_T & rhs ) [inline]
```

Definition at line 75 of file setdiff.h.

## 6.718.4 Member Function Documentation

### 6.718.4.1 get()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
bool Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get ( ) const [inline]
```

Definition at line 81 of file setdiff.h.

References `Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs_set`, and `Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs_set`.

Referenced by `Rcpp::setequal()`.

## 6.718.5 Member Data Documentation

### 6.718.5.1 lhs\_set

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
SET Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs_set [private]
```

Definition at line 94 of file setdiff.h.

Referenced by Rcpp::sugar::SetEqual< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::get().

### 6.718.5.2 rhs\_set

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
SET Rcpp::sugar::SetEqual< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs_set [private]
```

Definition at line 95 of file setdiff.h.

Referenced by Rcpp::sugar::SetEqual< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::get().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/setdiff.h

## 6.719 Rcpp::Shelter< T > Class Template Reference

```
#include <Shelter.h>
```

### Public Member Functions

- Shelter ()
- SEXP operator() (SEXP x)
- ~Shelter ()

### Private Member Functions

- Shelter (const Shelter &)
- Shelter & operator= (const Shelter &)

### Private Attributes

- int nprotected

## 6.719.1 Detailed Description

```
template<class T>
class Rcpp::Shelter< T >
```

Definition at line 24 of file Shelter.h.

## 6.719.2 Constructor & Destructor Documentation

### 6.719.2.1 Shelter() [1/2]

```
template<class T >
Rcpp::Shelter< T >::Shelter ( ) [inline]
```

Definition at line 26 of file Shelter.h.

### 6.719.2.2 ~Shelter()

```
template<class T >
Rcpp::Shelter< T >::~~Shelter ( ) [inline]
```

Definition at line 33 of file Shelter.h.

References Rcpp::Shelter< T >::nprotected, and Rcpp::Rcpp\_unprotect().

Here is the call graph for this function:



### 6.719.2.3 Shelter() [2/2]

```
template<class T >  
Rcpp::Shelter< T >::Shelter (   
    const Shelter< T > & ) [private]
```

## 6.719.3 Member Function Documentation

### 6.719.3.1 operator>()

```
template<class T >  
SEXP Rcpp::Shelter< T >::operator() (   
    SEXP x ) [inline]
```

Definition at line 28 of file Shelter.h.

References Rcpp::Shelter< T >::nprotected, and Rcpp::Rcpp\_protect().

Here is the call graph for this function:



### 6.719.3.2 operator=()

```
template<class T >  
Shelter& Rcpp::Shelter< T >::operator= (   
    const Shelter< T > & ) [private]
```

## 6.719.4 Member Data Documentation



### 6.719.4.1 nprotected

```
template<class T >
int Rcpp::Shelter< T >::nprotected [private]
```

Definition at line 39 of file Shelter.h.

Referenced by Rcpp::Shelter< T >::operator>(), and Rcpp::Shelter< T >::~~Shelter().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/protection/Shelter.h

## 6.720 Rcpp::Shield< T > Class Template Reference

```
#include <Shield.h>
```

### Public Member Functions

- [Shield](#) (SEXP t\_)
- [~Shield](#) ()
- [operator SEXP](#) () const

### Public Attributes

- [SEXP t](#)

### Private Member Functions

- [Shield](#) (const [Shield](#) &)
- [Shield](#) & [operator=](#) (const [Shield](#) &)

### 6.720.1 Detailed Description

```
template<typename T>
class Rcpp::Shield< T >
```

Definition at line 35 of file Shield.h.

### 6.720.2 Constructor & Destructor Documentation

### 6.720.2.1 Shield() [1/2]

```
template<typename T >
Rcpp::Shield< T >::Shield (
    SEXP t_ ) [inline]
```

Definition at line 37 of file Shield.h.

### 6.720.2.2 ~Shield()

```
template<typename T >
Rcpp::Shield< T >::~~Shield ( ) [inline]
```

Definition at line 38 of file Shield.h.

References Rcpp::Rcpp\_unprotect(), and Rcpp::Shield< T >::t.

Here is the call graph for this function:



### 6.720.2.3 Shield() [2/2]

```
template<typename T >
Rcpp::Shield< T >::Shield (
    const Shield< T > & ) [private]
```

## 6.720.3 Member Function Documentation

### 6.720.3.1 operator SEXP()

```
template<typename T >  
Rcpp::Shield< T >::operator SEXP ( ) const [inline]
```

Definition at line 42 of file Shield.h.

References Rcpp::Shield< T >::t.

### 6.720.3.2 operator=()

```
template<typename T >  
Shield& Rcpp::Shield< T >::operator= (  
    const Shield< T > & ) [private]
```

## 6.720.4 Member Data Documentation

### 6.720.4.1 t

```
template<typename T >  
SEXPRcpp::Shield< T >::t
```

Definition at line 43 of file Shield.h.

Referenced by Rcpp::Shield< T >::operator SEXP(), and Rcpp::Shield< T >::~~Shield().

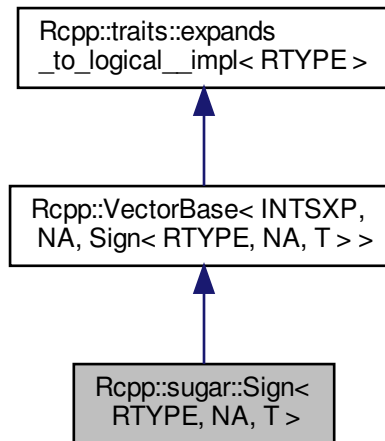
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/protection/Shield.h](#)

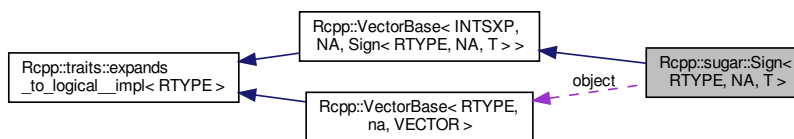
## 6.721 Rcpp::sugar::Sign< RTYPE, NA, T > Class Template Reference

```
#include <sign.h>
```

Inheritance diagram for Rcpp::sugar::Sign< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Sign< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`
- typedef int `r_import_type`

### Public Member Functions

- `Sign` (const `VEC_TYPE` &object\_)
- int `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const
- `operator SEXP` () const
- int `get` (R\_xlen\_t i) const

## Private Attributes

- const [VEC\\_TYPE](#) & object

### 6.721.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::Sign< RTYPE, NA, T >
```

Definition at line 48 of file sign.h.

### 6.721.2 Member Typedef Documentation

#### 6.721.2.1 r\_import\_type

```
template<int RTYPE, bool NA, typename T >  
typedef int Rcpp::sugar::Sign< RTYPE, NA, T >::r_import_type
```

Definition at line 52 of file sign.h.

#### 6.721.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Sign< RTYPE, NA, T >::STORAGE
```

Definition at line 51 of file sign.h.

#### 6.721.2.3 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Sign< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 50 of file sign.h.

### 6.721.3 Constructor & Destructor Documentation

### 6.721.3.1 Sign()

```
template<int RTYPE, bool NA, typename T >  
Rcpp::sugar::Sign< RTYPE, NA, T >::Sign (   
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 54 of file sign.h.

## 6.721.4 Member Function Documentation

### 6.721.4.1 get()

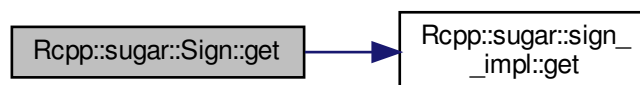
```
template<int RTYPE, bool NA, typename T >  
int Rcpp::sugar::Sign< RTYPE, NA, T >::get (   
    R_xlen_t i ) const [inline]
```

Definition at line 62 of file sign.h.

References Rcpp::sugar::sign\_\_impl< NA, RTYPE >::get().

Referenced by Rcpp::sugar::Sign< RTYPE, NA, T >::operator[]().

Here is the call graph for this function:



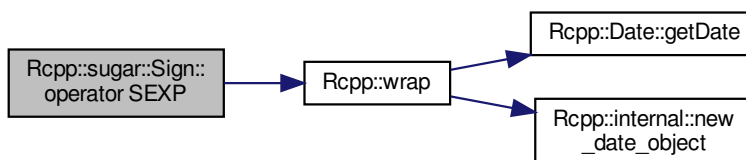
### 6.721.4.2 operator SEXP()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Sign< RTYPE, NA, T >::operator SEXP ( ) const [inline]
```

Definition at line 61 of file sign.h.

References Rcpp::wrap().

Here is the call graph for this function:



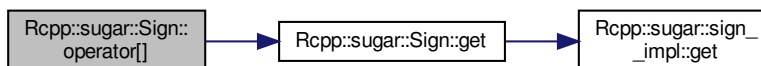
### 6.721.4.3 operator[]()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::Sign< RTYPE, NA, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 56 of file sign.h.

References Rcpp::sugar::Sign< RTYPE, NA, T >::get().

Here is the call graph for this function:



#### 6.721.4.4 size()

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Sign< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 59 of file sign.h.

### 6.721.5 Member Data Documentation

#### 6.721.5.1 object

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::sugar::Sign< RTYPE, NA, T >::object [private]
```

Definition at line 64 of file sign.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/sign.h](#)

## 6.722 Rcpp::sugar::sign\_\_impl< NA, RTYPE > Class Template Reference

```
#include <sign.h>
```

### Public Types

- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) STORAGE

### Static Public Member Functions

- static int [get](#) (STORAGE x)

#### 6.722.1 Detailed Description

```
template<bool NA, int RTYPE>  
class Rcpp::sugar::sign__impl< NA, RTYPE >
```

Definition at line 29 of file sign.h.



## 6.722.2 Member Typedef Documentation

### 6.722.2.1 STORAGE

```
template<bool NA, int RTYPE>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::sign__impl< NA, RTYPE >::STORAGE
```

Definition at line 31 of file sign.h.

## 6.722.3 Member Function Documentation

### 6.722.3.1 get()

```
template<bool NA, int RTYPE>
static int Rcpp::sugar::sign__impl< NA, RTYPE >::get (
    STORAGE x ) [inline], [static]
```

Definition at line 32 of file sign.h.

Referenced by Rcpp::sugar::Sign< RTYPE, NA, T >::get().

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/sign.h](#)

## 6.723 Rcpp::sugar::sign\_\_impl< false, RTYPE > Class Template Reference

```
#include <sign.h>
```

### Public Types

- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) STORAGE

### Static Public Member Functions

- static int [get](#) (STORAGE x)

### 6.723.1 Detailed Description

```
template<int RTYPE>
class Rcpp::sugar::sign__impl< false, RTYPE >
```

Definition at line 38 of file sign.h.

### 6.723.2 Member Typedef Documentation

#### 6.723.2.1 STORAGE

```
template<int RTYPE>
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::sign__impl< false, RTYPE >::STORAGE
```

Definition at line 40 of file sign.h.

### 6.723.3 Member Function Documentation

#### 6.723.3.1 get()

```
template<int RTYPE>
static int Rcpp::sugar::sign__impl< false, RTYPE >::get (
    STORAGE x ) [inline], [static]
```

Definition at line 41 of file sign.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/sign.h](#)

## 6.724 Rcpp::SignedConstructor< Class > Class Template Reference

```
#include <Module.h>
```

### Public Member Functions

- [SignedConstructor](#) (Constructor\_Base< Class > \*ctor\_, [ValidConstructor](#) valid\_, const char \*doc)
- int [nargs](#) ()
- void [signature](#) (std::string &buffer, const std::string &class\_name)

## Public Attributes

- [Constructor\\_Base< Class > \\* ctor](#)
- [ValidConstructor valid](#)
- `std::string docstring`

### 6.724.1 Detailed Description

```
template<typename Class>  
class Rcpp::SignedConstructor< Class >
```

Definition at line 143 of file Module.h.

### 6.724.2 Constructor & Destructor Documentation

#### 6.724.2.1 SignedConstructor()

```
template<typename Class >  
Rcpp::SignedConstructor< Class >::SignedConstructor (  
    Constructor_Base< Class > * ctor_,  
    ValidConstructor valid_,  
    const char * doc ) [inline]
```

Definition at line 146 of file Module.h.

### 6.724.3 Member Function Documentation

#### 6.724.3.1 nargs()

```
template<typename Class >  
int Rcpp::SignedConstructor< Class >::nargs ( ) [inline]
```

Definition at line 156 of file Module.h.

References [Rcpp::SignedConstructor< Class >::ctor](#).

Referenced by [Rcpp::S4\\_CppConstructor< Class >::S4\\_CppConstructor\(\)](#).

### 6.724.3.2 signature()

```
template<typename Class >
void Rcpp::SignedConstructor< Class >::signature (
    std::string & buffer,
    const std::string & class_name ) [inline]
```

Definition at line 157 of file Module.h.

References Rcpp::SignedConstructor< Class >::ctor.

Referenced by Rcpp::S4\_CppConstructor< Class >::S4\_CppConstructor().

## 6.724.4 Member Data Documentation

### 6.724.4.1 ctor

```
template<typename Class >
Constructor_Base<Class>* Rcpp::SignedConstructor< Class >::ctor
```

Definition at line 152 of file Module.h.

Referenced by Rcpp::SignedConstructor< Class >::nargs(), and Rcpp::SignedConstructor< Class >::signature().

### 6.724.4.2 docstring

```
template<typename Class >
std::string Rcpp::SignedConstructor< Class >::docstring
```

Definition at line 154 of file Module.h.

Referenced by Rcpp::S4\_CppConstructor< Class >::S4\_CppConstructor().

### 6.724.4.3 valid

```
template<typename Class >
ValidConstructor Rcpp::SignedConstructor< Class >::valid
```

Definition at line 153 of file Module.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

## 6.725 Rcpp::SignedFactory< Class > Class Template Reference

```
#include <Module.h>
```

### Public Member Functions

- [SignedFactory](#) (Factory\_Base< Class > \*fact\_, [ValidConstructor](#) valid\_, const char \*doc)
- int [nargs](#) ()
- void [signature](#) (std::string &buffer, const std::string &class\_name)

### Public Attributes

- Factory\_Base< Class > \* [fact](#)
- [ValidConstructor](#) [valid](#)
- std::string [docstring](#)

### 6.725.1 Detailed Description

```
template<typename Class >  
class Rcpp::SignedFactory< Class >
```

Definition at line 163 of file Module.h.

### 6.725.2 Constructor & Destructor Documentation

#### 6.725.2.1 SignedFactory()

```
template<typename Class >  
Rcpp::SignedFactory< Class >::SignedFactory (  
    Factory_Base< Class > * fact_,  
    ValidConstructor valid_,  
    const char * doc ) [inline]
```

Definition at line 166 of file Module.h.

### 6.725.3 Member Function Documentation

### 6.725.3.1 nargs()

```
template<typename Class >
int Rcpp::SignedFactory< Class >::nargs ( ) [inline]
```

Definition at line 176 of file Module.h.

References Rcpp::SignedFactory< Class >::fact.

### 6.725.3.2 signature()

```
template<typename Class >
void Rcpp::SignedFactory< Class >::signature (
    std::string & buffer,
    const std::string & class_name ) [inline]
```

Definition at line 177 of file Module.h.

References Rcpp::SignedFactory< Class >::fact.

## 6.725.4 Member Data Documentation

### 6.725.4.1 docstring

```
template<typename Class >
std::string Rcpp::SignedFactory< Class >::docstring
```

Definition at line 174 of file Module.h.

### 6.725.4.2 fact

```
template<typename Class >
Factory_Base<Class>* Rcpp::SignedFactory< Class >::fact
```

Definition at line 172 of file Module.h.

Referenced by Rcpp::SignedFactory< Class >::nargs(), and Rcpp::SignedFactory< Class >::signature().

## 6.725.4.3 valid

```
template<typename Class >
ValidConstructor Rcpp::SignedFactory< Class >::valid
```

Definition at line 173 of file Module.h.

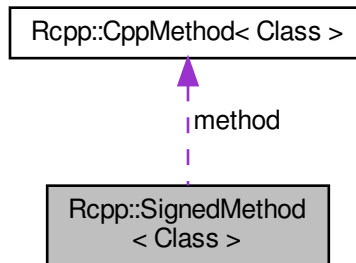
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

## 6.726 Rcpp::SignedMethod&lt; Class &gt; Class Template Reference

```
#include <Module.h>
```

Collaboration diagram for Rcpp::SignedMethod< Class >:



## Public Types

- typedef [CppMethod< Class >](#) [METHOD](#)

## Public Member Functions

- [SignedMethod](#) ([METHOD](#) \*m, [ValidMethod](#) valid\_, const char \*doc)
- int [nargs](#) ()
- bool [is\\_void](#) ()
- bool [is\\_const](#) ()
- void [signature](#) (std::string &s, const char \*name)

## Public Attributes

- [METHOD](#) \* [method](#)
- [ValidMethod](#) [valid](#)
- `std::string` [docstring](#)

### 6.726.1 Detailed Description

```
template<typename Class>  
class Rcpp::SignedMethod< Class >
```

Definition at line 184 of file Module.h.

### 6.726.2 Member Typedef Documentation

#### 6.726.2.1 METHOD

```
template<typename Class >  
typedef CppMethod<Class> Rcpp::SignedMethod< Class >::METHOD
```

Definition at line 186 of file Module.h.

### 6.726.3 Constructor & Destructor Documentation

#### 6.726.3.1 SignedMethod()

```
template<typename Class >  
Rcpp::SignedMethod< Class >::SignedMethod (  
    METHOD * m,  
    ValidMethod valid_,  
    const char * doc ) [inline]
```

Definition at line 187 of file Module.h.

### 6.726.4 Member Function Documentation



### 6.726.4.1 is\_const()

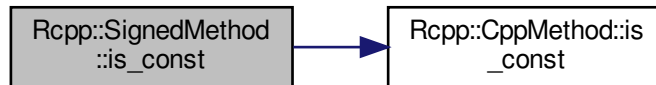
```
template<typename Class >  
bool Rcpp::SignedMethod< Class >::is_const ( ) [inline]
```

Definition at line 195 of file Module.h.

References Rcpp::CppMethod< Class >::is\_const(), and Rcpp::SignedMethod< Class >::method.

Referenced by Rcpp::S4\_CppOverloadedMethods< Class >::S4\_CppOverloadedMethods().

Here is the call graph for this function:



### 6.726.4.2 is\_void()

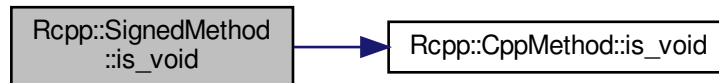
```
template<typename Class >  
bool Rcpp::SignedMethod< Class >::is_void ( ) [inline]
```

Definition at line 194 of file Module.h.

References Rcpp::CppMethod< Class >::is\_void(), and Rcpp::SignedMethod< Class >::method.

Referenced by Rcpp::S4\_CppOverloadedMethods< Class >::S4\_CppOverloadedMethods().

Here is the call graph for this function:



### 6.726.4.3 nargs()

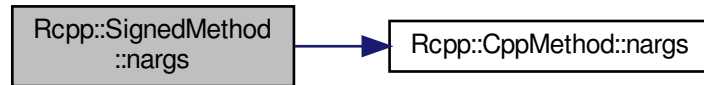
```
template<typename Class >
int Rcpp::SignedMethod< Class >::nargs ( ) [inline]
```

Definition at line 193 of file Module.h.

References Rcpp::SignedMethod< Class >::method, and Rcpp::CppMethod< Class >::nargs().

Referenced by Rcpp::S4\_CppOverloadedMethods< Class >::S4\_CppOverloadedMethods().

Here is the call graph for this function:



### 6.726.4.4 signature()

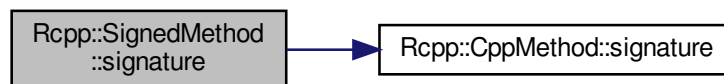
```
template<typename Class >
void Rcpp::SignedMethod< Class >::signature (
    std::string & s,
    const char * name ) [inline]
```

Definition at line 196 of file Module.h.

References Rcpp::SignedMethod< Class >::method, and Rcpp::CppMethod< Class >::signature().

Referenced by Rcpp::S4\_CppOverloadedMethods< Class >::S4\_CppOverloadedMethods().

Here is the call graph for this function:



## 6.726.5 Member Data Documentation

### 6.726.5.1 docstring

```
template<typename Class >  
std::string Rcpp::SignedMethod< Class >::docstring
```

Definition at line 191 of file Module.h.

Referenced by Rcpp::S4\_CppOverloadedMethods< Class >::S4\_CppOverloadedMethods().

### 6.726.5.2 method

```
template<typename Class >  
METHOD* Rcpp::SignedMethod< Class >::method
```

Definition at line 189 of file Module.h.

Referenced by Rcpp::SignedMethod< Class >::is\_const(), Rcpp::SignedMethod< Class >::is\_void(), Rcpp::SignedMethod< Class >::nargs(), and Rcpp::SignedMethod< Class >::signature().

### 6.726.5.3 valid

```
template<typename Class >  
ValidMethod Rcpp::SignedMethod< Class >::valid
```

Definition at line 190 of file Module.h.

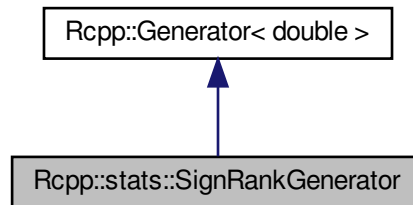
The documentation for this class was generated from the following file:

- inst/include/Rcpp/Module.h

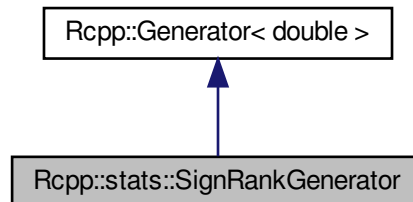
## 6.727 Rcpp::stats::SignRankGenerator Class Reference

```
#include <rsignrank.h>
```

Inheritance diagram for Rcpp::stats::SignRankGenerator:



Collaboration diagram for Rcpp::stats::SignRankGenerator:



### Public Member Functions

- [SignRankGenerator](#) (double nn\_)
- double [operator\(\)](#) () const

### Private Attributes

- double [nn](#)

## Additional Inherited Members

### 6.727.1 Detailed Description

Definition at line 29 of file rsignrank.h.

### 6.727.2 Constructor & Destructor Documentation

#### 6.727.2.1 SignRankGenerator()

```
Rcpp::stats::SignRankGenerator::SignRankGenerator (  
    double nn_ ) [inline]
```

Definition at line 31 of file rsignrank.h.

### 6.727.3 Member Function Documentation

#### 6.727.3.1 operator>()

```
double Rcpp::stats::SignRankGenerator::operator() ( ) const [inline]
```

Definition at line 32 of file rsignrank.h.

References nn.

### 6.727.4 Member Data Documentation

#### 6.727.4.1 nn

```
double Rcpp::stats::SignRankGenerator::nn [private]
```

Definition at line 34 of file rsignrank.h.

Referenced by operator>().

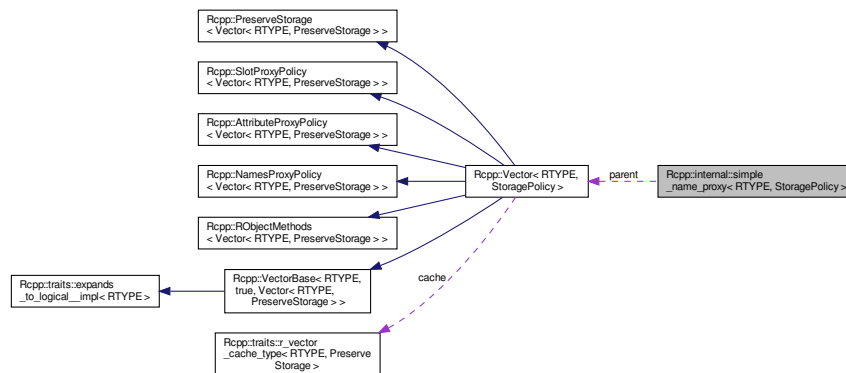
The documentation for this class was generated from the following file:

- inst/include/Rcpp/stats/random/rsignrank.h

## 6.728 Rcpp::internal::simple\_name\_proxy< RTYPE, StoragePolicy > Class Template Reference

```
#include <proxy.h>
```

Collaboration diagram for Rcpp::internal::simple\_name\_proxy< RTYPE, StoragePolicy >:



### Public Types

- typedef `::Rcpp::Vector< RTYPE, StoragePolicy >` `VECTOR`
- typedef `::Rcpp::traits::storage_type< RTYPE >::type` `CTYPE`

### Public Member Functions

- `simple_name_proxy` (`VECTOR &v`, `const std::string &name_`)
- `simple_name_proxy` (`const simple_name_proxy &other`)
- `~simple_name_proxy` ()
- `simple_name_proxy & operator=` (`CTYPE rhs`)
- `simple_name_proxy & operator=` (`const simple_name_proxy &other`)
- `template<typename T >`  
`simple_name_proxy & operator=` (`const T &rhs`)
- `operator CTYPE` () `const`
- `operator SEXP` () `const`

### Private Member Functions

- `void set` (`CTYPE rhs`)
- `CTYPE get` () `const`

## Private Attributes

- [VECTOR](#) & [parent](#)
- `std::string` [name](#)

### 6.728.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>  
class Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >
```

Definition at line 29 of file proxy.h.

### 6.728.2 Member Typedef Documentation

#### 6.728.2.1 CTYPE

```
template<int RTYPE, template< class > class StoragePolicy>  
typedef ::Rcpp::traits::storage_type<RTYPE>::type Rcpp::internal::simple_name_proxy< RTYPE, Storage↵  
Policy >::CTYPE
```

Definition at line 32 of file proxy.h.

#### 6.728.2.2 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>  
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::simple_name_proxy< RTYPE, Storage↵  
Policy >::VECTOR
```

Definition at line 31 of file proxy.h.

### 6.728.3 Constructor & Destructor Documentation

**6.728.3.1 simple\_name\_proxy() [1/2]**

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::simple_name_proxy (
    VECTOR & v,
    const std::string & name_ ) [inline]
```

Definition at line 34 of file proxy.h.

**6.728.3.2 simple\_name\_proxy() [2/2]**

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::simple_name_proxy (
    const simple_name_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 36 of file proxy.h.

**6.728.3.3 ~simple\_name\_proxy()**

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::~simple_name_proxy ( ) [inline]
```

Definition at line 38 of file proxy.h.

**6.728.4 Member Function Documentation****6.728.4.1 get()**

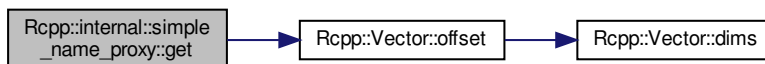
```
template<int RTYPE, template< class > class StoragePolicy>
CTYPE Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::get ( ) const [inline], [private]
```

Definition at line 78 of file proxy.h.

References `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::name`, `Rcpp::Vector< RTYPE, StoragePolicy >::offset()`, and `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::parent`.

Referenced by `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator CTYPE()`, `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator SEXP()`, and `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator=()`.

Here is the call graph for this function:





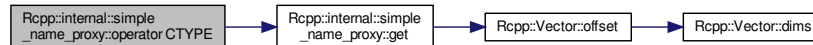
### 6.728.4.2 operator CTYPE()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator CTYPE ( ) const [inline]
```

Definition at line 57 of file proxy.h.

References Rcpp::internal::simple\_name\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



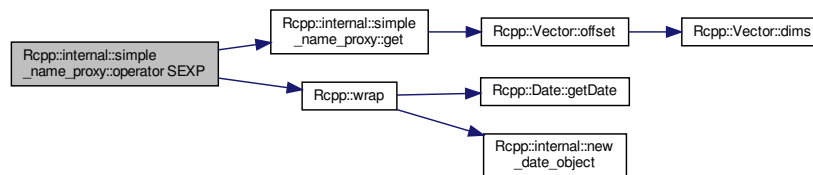
### 6.728.4.3 operator SEXP()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const [inline]
```

Definition at line 62 of file proxy.h.

References Rcpp::internal::simple\_name\_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::wrap().

Here is the call graph for this function:



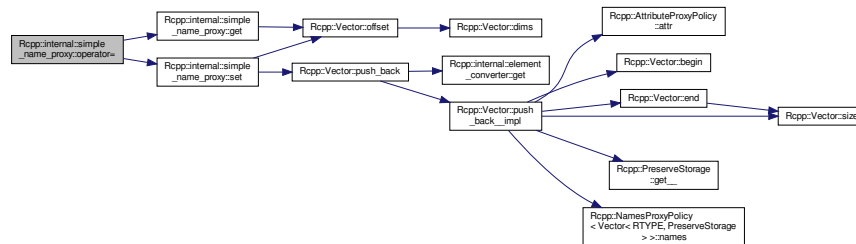
#### 6.728.4.4 operator=() [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
simple_name_proxy& Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator= (
    const simple_name_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 44 of file proxy.h.

References `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::get()`, and `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set()`.

Here is the call graph for this function:



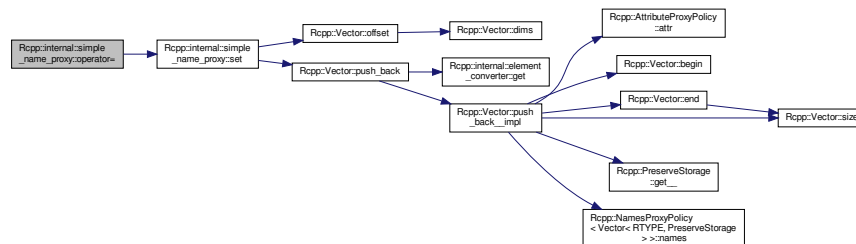
#### 6.728.4.5 operator=() [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
simple_name_proxy& Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator= (
    const T & rhs ) [inline]
```

Definition at line 50 of file proxy.h.

References `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set()`.

Here is the call graph for this function:



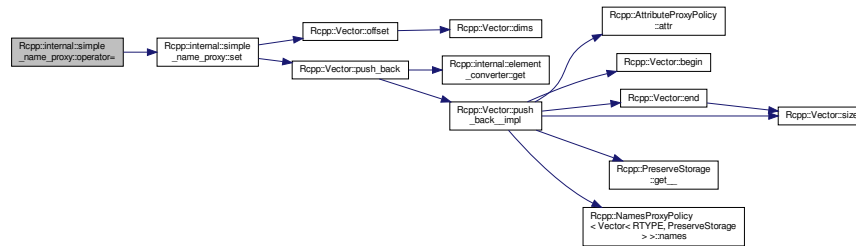
## 6.728.4.6 operator=() [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
simple_name_proxy& Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::operator= (
    CTYPE rhs ) [inline]
```

Definition at line 40 of file proxy.h.

References Rcpp::internal::simple\_name\_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



## 6.728.4.7 set()

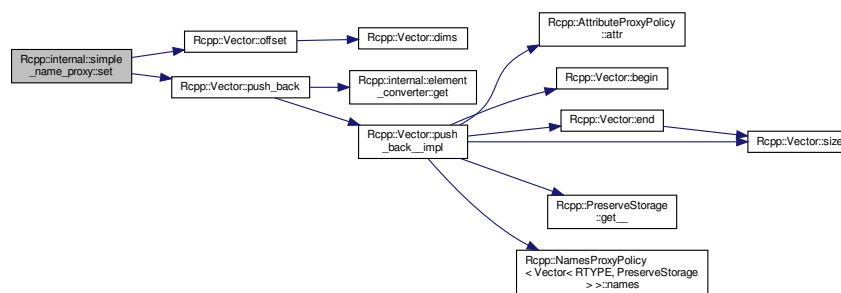
```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set (
    CTYPE rhs ) [inline], [private]
```

Definition at line 69 of file proxy.h.

References Rcpp::internal::simple\_name\_proxy< RTYPE, StoragePolicy >::name, Rcpp::Vector< RTYPE, StoragePolicy >::offset(), Rcpp::internal::simple\_name\_proxy< RTYPE, StoragePolicy >::parent, and Rcpp::Vector< RTYPE, StoragePolicy >::push\_back().

Referenced by Rcpp::internal::simple\_name\_proxy< RTYPE, StoragePolicy >::operator=().

Here is the call graph for this function:



## 6.728.5 Member Data Documentation

### 6.728.5.1 name

```
template<int RTYPE, template< class > class StoragePolicy>  
std::string Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::name [private]
```

Definition at line 68 of file proxy.h.

Referenced by `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::get()`, and `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set()`.

### 6.728.5.2 parent

```
template<int RTYPE, template< class > class StoragePolicy>  
VECTOR& Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::parent [private]
```

Definition at line 67 of file proxy.h.

Referenced by `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::get()`, and `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set()`.

The documentation for this class was generated from the following files:

- [inst/include/Rcpp/vector/00\\_forward\\_proxy.h](#)
- [inst/include/Rcpp/vector/proxy.h](#)

## 6.729 Rcpp::SingleLogicalResult< NA, T > Class Template Reference

### 6.729.1 Detailed Description

```
template<bool NA, typename T>  
class Rcpp::SingleLogicalResult< NA, T >
```

Definition at line 40 of file Vector.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Vector.h](#)

## 6.730 Rcpp::sugar::SingleLogicalResult< NA, T > Class Template Reference

```
#include <SingleLogicalResult.h>
```

### Public Member Functions

- [SingleLogicalResult](#) ()
- void [apply](#) ()
- bool [is\\_true](#) ()
- bool [is\\_false](#) ()
- bool [is\\_na](#) ()
- [operator SEXP](#) ()
- [operator bool](#) ()
- int [size](#) ()
- int [get](#) ()
- SEXP [get\\_sexp](#) ()

### Static Public Attributes

- static const int [UNRESOLVED](#) = -5

### Protected Member Functions

- void [set](#) (int x)
- void [reset](#) ()
- void [set\\_true](#) ()
- void [set\\_false](#) ()
- void [set\\_na](#) ()
- bool [is\\_unresolved](#) ()

### Protected Attributes

- int [result](#)

### 6.730.1 Detailed Description

```
template<bool NA, typename T>  
class Rcpp::sugar::SingleLogicalResult< NA, T >
```

Definition at line 42 of file SingleLogicalResult.h.

## 6.730.2 Constructor & Destructor Documentation

### 6.730.2.1 SingleLogicalResult()

```
template<bool NA, typename T >
Rcpp::sugar::SingleLogicalResult< NA, T >::SingleLogicalResult ( ) [inline]
```

Definition at line 46 of file SingleLogicalResult.h.

## 6.730.3 Member Function Documentation

### 6.730.3.1 apply()

```
template<bool NA, typename T >
void Rcpp::sugar::SingleLogicalResult< NA, T >::apply ( ) [inline]
```

Definition at line 48 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::UNRESOLVED.

Referenced by Rcpp::sugar::SingleLogicalResult< NA, T >::get(), Rcpp::sugar::SingleLogicalResult< NA, T >::get←\_sexp(), Rcpp::sugar::SingleLogicalResult< NA, T >::is\_false(), Rcpp::sugar::SingleLogicalResult< NA, T >::is\_na(), and Rcpp::sugar::SingleLogicalResult< NA, T >::is\_true().

### 6.730.3.2 get()

```
template<bool NA, typename T >
int Rcpp::sugar::SingleLogicalResult< NA, T >::get ( ) [inline]
```

Definition at line 81 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::apply(), and Rcpp::sugar::SingleLogicalResult< NA, T >←::result.

Referenced by Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >←::apply(), Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, false, RHS\_T >::apply(), Rcpp::sugar::And\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, RHS\_NA, RHS\_T >::apply(), Rcpp←::sugar::And\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, false, RHS\_T >::apply(), Rcpp::sugar::And\_←SingleLogicalResult\_bool< LHS\_NA, LHS\_T >::apply(), Rcpp::sugar::Negate\_SingleLogicalResult< NA, T >::apply(),

Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::apply(), Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< LHS\_NA, LHS\_T, false, RHS\_T >::apply(), Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, RHS\_NA, RHS\_T >::apply(), Rcpp::sugar::Or\_SingleLogicalResult\_SingleLogicalResult< false, LHS\_T, false, RHS\_T >::apply(), and Rcpp::sugar::Or\_SingleLogicalResult\_bool< LHS\_NA, LHS\_T >::apply().

Here is the call graph for this function:



### 6.730.3.3 get\_sexp()

```

template<bool NA, typename T >
SEXP Rcpp::sugar::SingleLogicalResult< NA, T >::get_sexp ( ) [inline]
  
```

Definition at line 86 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::apply().

Referenced by Rcpp::sugar::SingleLogicalResult< NA, T >::operator SEXP().

Here is the call graph for this function:



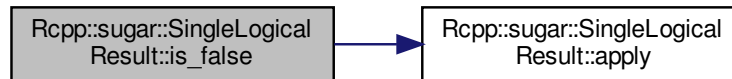
### 6.730.3.4 is\_false()

```
template<bool NA, typename T >
bool Rcpp::sugar::SingleLogicalResult< NA, T >::is_false ( ) [inline]
```

Definition at line 59 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::apply().

Here is the call graph for this function:



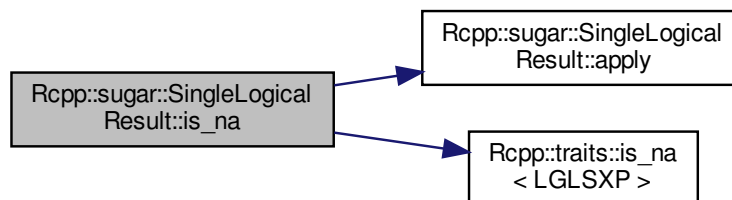
### 6.730.3.5 is\_na()

```
template<bool NA, typename T >
bool Rcpp::sugar::SingleLogicalResult< NA, T >::is_na ( ) [inline]
```

Definition at line 64 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::apply(), and Rcpp::traits::is\_na< LGLSXP >().

Here is the call graph for this function:





**6.730.3.6 is\_true()**

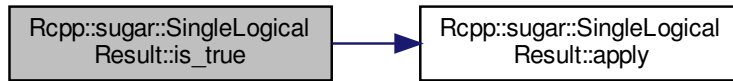
```
template<bool NA, typename T >
bool Rcpp::sugar::SingleLogicalResult< NA, T >::is_true ( ) [inline]
```

Definition at line 54 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::apply().

Referenced by Rcpp::sugar::SingleLogicalResult< NA, T >::operator bool().

Here is the call graph for this function:

**6.730.3.7 is\_unresolved()**

```
template<bool NA, typename T >
bool Rcpp::sugar::SingleLogicalResult< NA, T >::is_unresolved ( ) [inline], [protected]
```

Definition at line 98 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::UNRESOLVED.

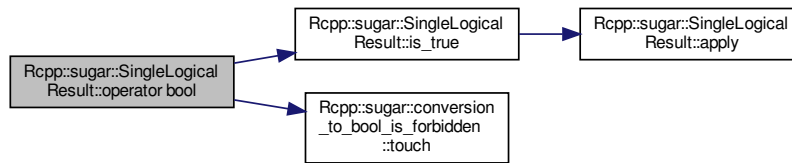
**6.730.3.8 operator bool()**

```
template<bool NA, typename T >
Rcpp::sugar::SingleLogicalResult< NA, T >::operator bool ( ) [inline]
```

Definition at line 73 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::is\_true(), and Rcpp::sugar::conversion\_to\_bool\_is\_forbidden< x >::touch().

Here is the call graph for this function:



### 6.730.3.9 operator SEXP()

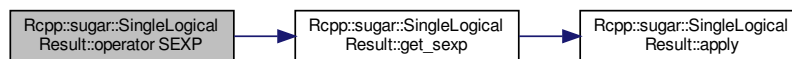
```

template<bool NA, typename T >
Rcpp::sugar::SingleLogicalResult< NA, T >::operator SEXP ( ) [inline]
  
```

Definition at line 69 of file SingleLogicalResult.h.

References `Rcpp::sugar::SingleLogicalResult< NA, T >::get_sexp()`.

Here is the call graph for this function:



### 6.730.3.10 reset()

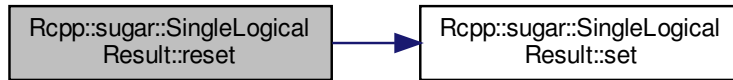
```

template<bool NA, typename T >
void Rcpp::sugar::SingleLogicalResult< NA, T >::reset ( ) [inline], [protected]
  
```

Definition at line 94 of file SingleLogicalResult.h.

References `Rcpp::sugar::SingleLogicalResult< NA, T >::set()`, and `Rcpp::sugar::SingleLogicalResult< NA, T >::←UNRESOLVED`.

Here is the call graph for this function:



### 6.730.3.11 set()

```

template<bool NA, typename T >
void Rcpp::sugar::SingleLogicalResult< NA, T >::set (
    int x ) [inline], [protected]
  
```

Definition at line 93 of file SingleLogicalResult.h.

Referenced by `Rcpp::sugar::Or_SingleLogicalResult_bool< LHS_NA, LHS_T >::apply()`, `Rcpp::sugar::SingleLogicalResult< NA, T >::reset()`, `Rcpp::sugar::SingleLogicalResult< NA, T >::set_false()`, `Rcpp::sugar::SingleLogicalResult< NA, T >::set_na()`, and `Rcpp::sugar::SingleLogicalResult< NA, T >::set_true()`.

### 6.730.3.12 set\_false()

```

template<bool NA, typename T >
void Rcpp::sugar::SingleLogicalResult< NA, T >::set_false ( ) [inline], [protected]
  
```

Definition at line 96 of file SingleLogicalResult.h.

References `Rcpp::sugar::SingleLogicalResult< NA, T >::set()`.

Here is the call graph for this function:



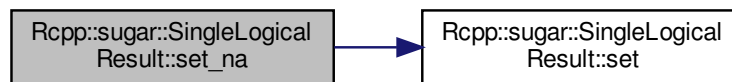
### 6.730.3.13 set\_na()

```
template<bool NA, typename T >  
void Rcpp::sugar::SingleLogicalResult< NA, T >::set_na ( ) [inline], [protected]
```

Definition at line 97 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::set().

Here is the call graph for this function:



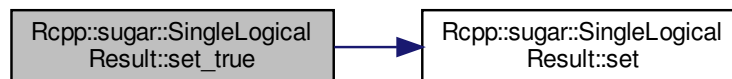
### 6.730.3.14 set\_true()

```
template<bool NA, typename T >  
void Rcpp::sugar::SingleLogicalResult< NA, T >::set_true ( ) [inline], [protected]
```

Definition at line 95 of file SingleLogicalResult.h.

References Rcpp::sugar::SingleLogicalResult< NA, T >::set().

Here is the call graph for this function:



### 6.730.3.15 size()

```
template<bool NA, typename T >
int Rcpp::sugar::SingleLogicalResult< NA, T >::size ( ) [inline]
```

Definition at line 79 of file SingleLogicalResult.h.

## 6.730.4 Member Data Documentation

### 6.730.4.1 result

```
template<bool NA, typename T >
int Rcpp::sugar::SingleLogicalResult< NA, T >::result [protected]
```

Definition at line 92 of file SingleLogicalResult.h.

Referenced by Rcpp::sugar::SingleLogicalResult< NA, T >::get().

### 6.730.4.2 UNRESOLVED

```
template<bool NA, typename T >
const int Rcpp::sugar::SingleLogicalResult< NA, T >::UNRESOLVED = -5 [static]
```

Definition at line 44 of file SingleLogicalResult.h.

Referenced by Rcpp::sugar::SingleLogicalResult< NA, T >::apply(), Rcpp::sugar::SingleLogicalResult< NA, T >::is←\_unresolved(), and Rcpp::sugar::SingleLogicalResult< NA, T >::reset().

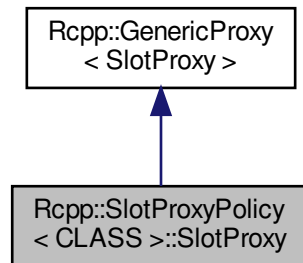
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/logical/SingleLogicalResult.h

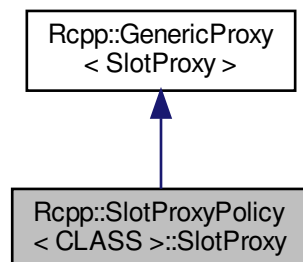
## 6.731 Rcpp::SlotProxyPolicy< CLASS >::SlotProxy Class Reference

```
#include <SlotProxy.h>
```

Inheritance diagram for Rcpp::SlotProxyPolicy< CLASS >::SlotProxy:



Collaboration diagram for Rcpp::SlotProxyPolicy< CLASS >::SlotProxy:



### Public Member Functions

- [SlotProxy](#) (CLASS &v, const std::string &name)
- [SlotProxy](#) & [operator=](#) (const [SlotProxy](#) &rhs)
- template<typename T >  
[SlotProxy](#) & [operator=](#) (const T &rhs)
- template<typename T >  
[operator T](#) () const
- [operator SEXP](#) () const
- template<typename T >  
[SlotProxyPolicy](#)< CLASS >::[SlotProxy](#) & [operator=](#) (const T &rhs)

## Private Member Functions

- SEXP [get](#) () const
- void [set](#) (SEXP x)

## Private Attributes

- CLASS & [parent](#)
- SEXP [slot\\_name](#)

### 6.731.1 Detailed Description

```
template<typename CLASS>
class Rcpp::SlotProxyPolicy< CLASS >::SlotProxy
```

Definition at line 27 of file SlotProxy.h.

### 6.731.2 Constructor & Destructor Documentation

#### 6.731.2.1 SlotProxy()

```
template<typename CLASS >
Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::SlotProxy (
    CLASS & v,
    const std::string & name ) [inline]
```

Definition at line 29 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::slot\_name.

### 6.731.3 Member Function Documentation

#### 6.731.3.1 get()

```
template<typename CLASS >
SEXP Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::get ( ) const [inline], [private]
```

Definition at line 51 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::parent, and Rcpp::SlotProxyPolicy< CLASS >::SlotProxy↔::slot\_name.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator SEXP(), and Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator=().

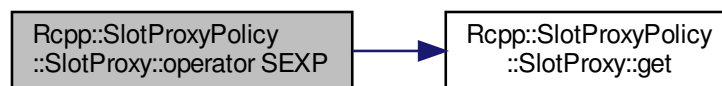
### 6.731.3.2 operator SEXP()

```
template<typename CLASS >  
Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator SEXP ( ) const [inline]
```

Definition at line 43 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::get().

Here is the call graph for this function:



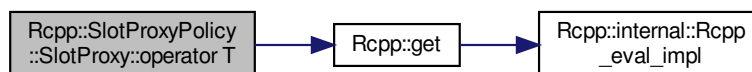
### 6.731.3.3 operator T()

```
template<typename CLASS >  
template<typename T >  
Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator T
```

Definition at line 91 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:





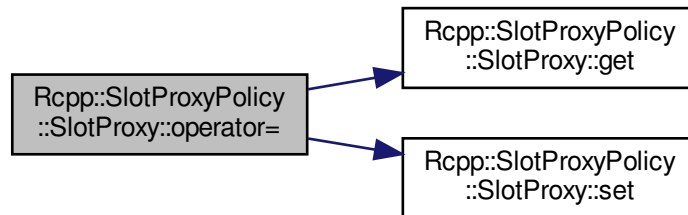
**6.731.3.4 operator=()** [1/3]

```
template<typename CLASS >
SlotProxy& Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator= (
    const SlotProxy & rhs ) [inline]
```

Definition at line 35 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::get(), and Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::set().

Here is the call graph for this function:

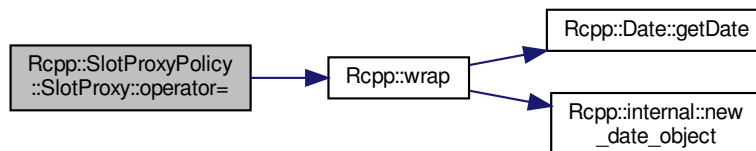
**6.731.3.5 operator=()** [2/3]

```
template<typename CLASS >
template<typename T >
SlotProxyPolicy<CLASS>::SlotProxy& Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator= (
    const T & rhs )
```

Definition at line 84 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:



### 6.731.3.6 operator=() [3/3]

```
template<typename CLASS >
template<typename T >
SlotProxy& Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator= (
    const T & rhs )
```

### 6.731.3.7 set()

```
template<typename CLASS >
void Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::set (
    SEXP x ) [inline], [private]
```

Definition at line 54 of file SlotProxy.h.

References Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::parent, and Rcpp::SlotProxyPolicy< CLASS >::SlotProxy↔::slot\_name.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::operator=().

## 6.731.4 Member Data Documentation

### 6.731.4.1 parent

```
template<typename CLASS >
CLASS& Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::parent [private]
```

Definition at line 48 of file SlotProxy.h.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::get(), and Rcpp::SlotProxyPolicy< CLASS >::Slot↔Proxy::set().

### 6.731.4.2 slot\_name

```
template<typename CLASS >
SEXP Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::slot_name [private]
```

Definition at line 49 of file SlotProxy.h.

Referenced by Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::get(), Rcpp::SlotProxyPolicy< CLASS >::SlotProxy↔::set(), and Rcpp::SlotProxyPolicy< CLASS >::SlotProxy::SlotProxy().

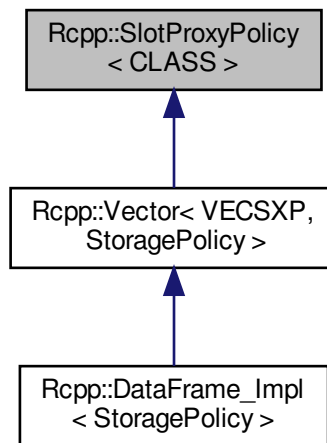
The documentation for this class was generated from the following files:

- inst/include/Rcpp/proxy/SlotProxy.h
- inst/include/Rcpp/api/meat/proxy.h

## 6.732 Rcpp::SlotProxyPolicy< CLASS > Class Template Reference

```
#include <SlotProxy.h>
```

Inheritance diagram for Rcpp::SlotProxyPolicy< CLASS >:



### Classes

- class [const\\_SlotProxy](#)
- class [SlotProxy](#)

### Public Member Functions

- [SlotProxy slot](#) (const std::string &name)
- [const\\_SlotProxy slot](#) (const std::string &name) const
- bool [hasSlot](#) (const std::string &name) const

### 6.732.1 Detailed Description

```
template<typename CLASS>  
class Rcpp::SlotProxyPolicy< CLASS >
```

Definition at line 24 of file SlotProxy.h.

## 6.732.2 Member Function Documentation

### 6.732.2.1 hasSlot()

```
template<typename CLASS >
bool Rcpp::SlotProxyPolicy< CLASS >::hasSlot (
    const std::string & name ) const [inline]
```

Definition at line 94 of file SlotProxy.h.

### 6.732.2.2 slot() [1/2]

```
template<typename CLASS >
SlotProxy Rcpp::SlotProxyPolicy< CLASS >::slot (
    const std::string & name ) [inline]
```

Definition at line 83 of file SlotProxy.h.

### 6.732.2.3 slot() [2/2]

```
template<typename CLASS >
const SlotProxy Rcpp::SlotProxyPolicy< CLASS >::slot (
    const std::string & name ) const [inline]
```

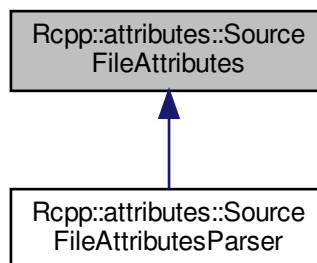
Definition at line 88 of file SlotProxy.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/proxy/SlotProxy.h

## 6.733 Rcpp::attributes::SourceFileAttributes Class Reference

Inheritance diagram for Rcpp::attributes::SourceFileAttributes:



## Public Types

- typedef std::vector< [Attribute](#) >::const\_iterator const\_iterator

## Public Member Functions

- virtual [~SourceFileAttributes](#) ()
- virtual const std::string & [sourceFile](#) () const =0
- virtual bool [hasInterface](#) (const std::string &name) const =0
- virtual [const\\_iterator begin](#) () const =0
- virtual [const\\_iterator end](#) () const =0
- virtual const std::vector< std::string > & [modules](#) () const =0
- virtual const std::vector< std::vector< std::string > > & [roxygenChunks](#) () const =0
- virtual bool [hasGeneratorOutput](#) () const =0
- virtual bool [hasPackageInit](#) () const =0

### 6.733.1 Detailed Description

Definition at line 436 of file attributes.cpp.

### 6.733.2 Member Typedef Documentation

#### 6.733.2.1 const\_iterator

```
typedef std::vector<Attribute>::const_iterator Rcpp::attributes::SourceFileAttributes::const_iterator
```

Definition at line 443 of file attributes.cpp.

### 6.733.3 Constructor & Destructor Documentation

#### 6.733.3.1 ~SourceFileAttributes()

```
virtual Rcpp::attributes::SourceFileAttributes::~SourceFileAttributes ( ) [inline], [virtual]
```

Definition at line 439 of file attributes.cpp.

## 6.733.4 Member Function Documentation

### 6.733.4.1 begin()

```
virtual const_iterator Rcpp::attributes::SourceFileAttributes::begin ( ) const [pure virtual]
```

Implemented in [Rcpp::attributes::SourceFileAttributesParser](#).

Referenced by [Rcpp::attributes::CppExportsGenerator::doWriteFunctions\(\)](#), [Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions\(\)](#), [Rcpp::attributes::REExportsGenerator::doWriteFunctions\(\)](#), and [Rcpp::attributes::generateCpp\(\)](#).

### 6.733.4.2 end()

```
virtual const_iterator Rcpp::attributes::SourceFileAttributes::end ( ) const [pure virtual]
```

Implemented in [Rcpp::attributes::SourceFileAttributesParser](#).

Referenced by [Rcpp::attributes::CppExportsGenerator::doWriteFunctions\(\)](#), [Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions\(\)](#), [Rcpp::attributes::REExportsGenerator::doWriteFunctions\(\)](#), and [Rcpp::attributes::generateCpp\(\)](#).

### 6.733.4.3 hasGeneratorOutput()

```
virtual bool Rcpp::attributes::SourceFileAttributes::hasGeneratorOutput ( ) const [pure virtual]
```

Implemented in [Rcpp::attributes::SourceFileAttributesParser](#).

### 6.733.4.4 hasInterface()

```
virtual bool Rcpp::attributes::SourceFileAttributes::hasInterface (
    const std::string & name ) const [pure virtual]
```

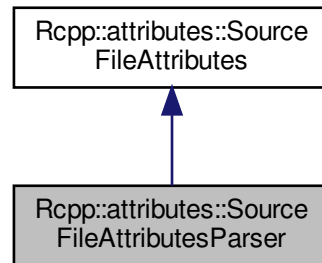
Implemented in [Rcpp::attributes::SourceFileAttributesParser](#).

Referenced by [Rcpp::attributes::CppExportsGenerator::doWriteFunctions\(\)](#), [Rcpp::attributes::CppExportsIncludeGenerator::doWriteFunctions\(\)](#), [Rcpp::attributes::REExportsGenerator::doWriteFunctions\(\)](#), and [Rcpp::attributes::ExportsGenerator::writeFunctions\(\)](#).

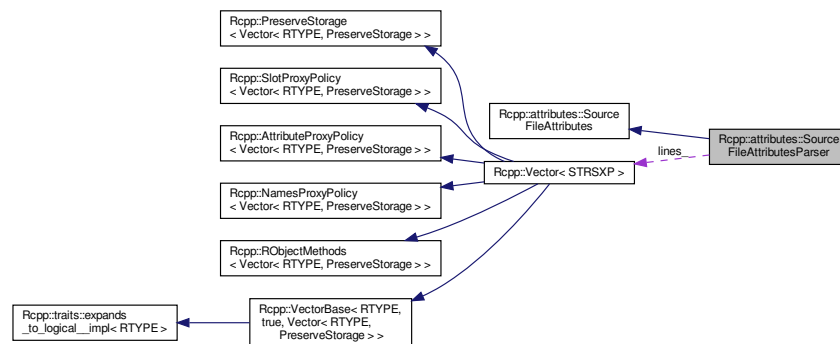


## 6.734 Rcpp::attributes::SourceFileAttributesParser Class Reference

Inheritance diagram for Rcpp::attributes::SourceFileAttributesParser:



Collaboration diagram for Rcpp::attributes::SourceFileAttributesParser:



### Public Member Functions

- [SourceFileAttributesParser](#) (const std::string &sourceFile, const std::string &packageFile, bool parse↔ Dependencies)
- virtual const std::string & [sourceFile](#) () const
- virtual [const\\_iterator begin](#) () const
- virtual [const\\_iterator end](#) () const
- virtual const std::vector< std::string > & [modules](#) () const
- virtual const std::vector< std::vector< std::string > > & [roxygenChunks](#) () const
- virtual bool [hasGeneratorOutput](#) () const
- virtual bool [hasInterface](#) (const std::string &name) const
- bool [hasPackageInit](#) () const
- const std::vector< std::string > & [embeddedR](#) () const
- const std::vector< [FileInfo](#) > & [sourceDependencies](#) () const



## Private Member Functions

- [SourceFileAttributesParser](#) (const [SourceFileAttributesParser](#) &)
- [SourceFileAttributesParser](#) & [operator=](#) (const [SourceFileAttributesParser](#) &)
- [Attribute](#) [parseAttribute](#) (const std::vector< std::string > &[match](#), int lineNumber)
- std::vector< [Param](#) > [parseParameters](#) (const std::string &input)
- [Function](#) [parseFunction](#) (size\_t lineNumber)
- std::string [parseSignature](#) (size\_t lineNumber)
- std::vector< std::string > [parseArguments](#) (const std::string &argText)
- [Type](#) [parseType](#) (const std::string &text)
- bool [isKnownAttribute](#) (const std::string &name) const
- void [attributeWarning](#) (const std::string &[message](#), const std::string &attribute, size\_t lineNumber)
- void [attributeWarning](#) (const std::string &[message](#), size\_t lineNumber)
- void [rcppExportWarning](#) (const std::string &[message](#), size\_t lineNumber)
- void [rcppExportNoFunctionFoundWarning](#) (size\_t lineNumber)
- void [rcppExportInvalidParameterWarning](#) (const std::string &param, size\_t lineNumber)
- void [rcppInterfacesWarning](#) (const std::string &[message](#), size\_t lineNumber)

## Private Attributes

- std::string [sourceFile\\_](#)
- [CharacterVector](#) [lines\\_](#)
- std::vector< [Attribute](#) > [attributes\\_](#)
- std::vector< std::string > [modules\\_](#)
- bool [hasPackageInit\\_](#)
- std::vector< std::string > [embeddedR\\_](#)
- std::vector< [FileInfo](#) > [sourceDependencies\\_](#)
- std::vector< std::vector< std::string > > [roxygenChunks\\_](#)
- std::vector< std::string > [roxygenBuffer\\_](#)

## Additional Inherited Members

### 6.734.1 Detailed Description

Definition at line 486 of file attributes.cpp.

### 6.734.2 Constructor & Destructor Documentation

#### 6.734.2.1 SourceFileAttributesParser() [1/2]

```
Rcpp::attributes::SourceFileAttributesParser::SourceFileAttributesParser (
    const std::string & sourceFile,
    const std::string & packageFile,
    bool parseDependencies ) [explicit]
```

### 6.734.2.2 SourceFileAttributesParser() [2/2]

```
Rcpp::attributes::SourceFileAttributesParser::SourceFileAttributesParser (
    const SourceFileAttributesParser & ) [private]
```

## 6.734.3 Member Function Documentation

### 6.734.3.1 attributeWarning() [1/2]

```
void Rcpp::attributes::SourceFileAttributesParser::attributeWarning (
    const std::string & message,
    const std::string & attribute,
    size_t lineNumber ) [private]
```

### 6.734.3.2 attributeWarning() [2/2]

```
void Rcpp::attributes::SourceFileAttributesParser::attributeWarning (
    const std::string & message,
    size_t lineNumber ) [private]
```

### 6.734.3.3 begin()

```
virtual const\_iterator Rcpp::attributes::SourceFileAttributesParser::begin ( ) const [inline],
[virtual]
```

Implements [Rcpp::attributes::SourceFileAttributes](#).

Definition at line 502 of file `attributes.cpp`.

References `attributes_`.

Referenced by `compileAttributes()`, and `hasInterface()`.

#### 6.734.3.4 embeddedR()

```
const std::vector<std::string>& Rcpp::attributes::SourceFileAttributesParser::embeddedR ( ) const  
[inline]
```

Definition at line 542 of file attributes.cpp.

References [embeddedR\\_](#).

#### 6.734.3.5 end()

```
virtual const\_iterator Rcpp::attributes::SourceFileAttributesParser::end ( ) const [inline],  
[virtual]
```

Implements [Rcpp::attributes::SourceFileAttributes](#).

Definition at line 503 of file attributes.cpp.

References [attributes\\_](#).

Referenced by [compileAttributes\(\)](#), and [hasInterface\(\)](#).

#### 6.734.3.6 hasGeneratorOutput()

```
virtual bool Rcpp::attributes::SourceFileAttributesParser::hasGeneratorOutput ( ) const [inline],  
[virtual]
```

Implements [Rcpp::attributes::SourceFileAttributes](#).

Definition at line 514 of file attributes.cpp.

References [attributes\\_](#), [modules\\_](#), and [roxygenChunks\\_](#).

Referenced by [compileAttributes\(\)](#).

### 6.734.3.7 hasInterface()

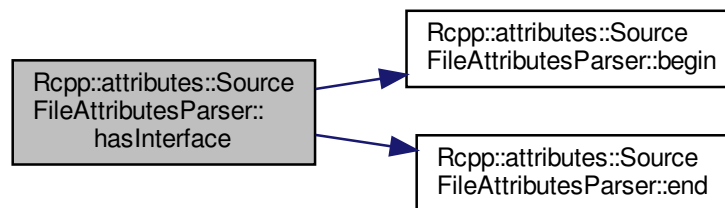
```
virtual bool Rcpp::attributes::SourceFileAttributesParser::hasInterface (
    const std::string & name ) const [inline], [virtual]
```

Implements [Rcpp::attributes::SourceFileAttributes](#).

Definition at line 521 of file attributes.cpp.

References [begin\(\)](#), [end\(\)](#), [Rcpp::attributes::kInterfaceR](#), and [Rcpp::attributes::kInterfacesAttribute](#).

Here is the call graph for this function:



### 6.734.3.8 hasPackageInit()

```
bool Rcpp::attributes::SourceFileAttributesParser::hasPackageInit ( ) const [inline], [virtual]
```

Implements [Rcpp::attributes::SourceFileAttributes](#).

Definition at line 537 of file attributes.cpp.

References [hasPackageInit\\_](#).

Referenced by [compileAttributes\(\)](#).

### 6.734.3.9 isKnownAttribute()

```
bool Rcpp::attributes::SourceFileAttributesParser::isKnownAttribute (
    const std::string & name ) const [private]
```

### 6.734.3.10 modules()

```
virtual const std::vector<std::string>& Rcpp::attributes::SourceFileAttributesParser::modules ( )  
const [inline], [virtual]
```

Implements [Rcpp::attributes::SourceFileAttributes](#).

Definition at line 505 of file attributes.cpp.

References [modules\\_](#).

### 6.734.3.11 operator=()

```
SourceFileAttributesParser& Rcpp::attributes::SourceFileAttributesParser::operator= (  
    const SourceFileAttributesParser & ) [private]
```

### 6.734.3.12 parseArguments()

```
std::vector<std::string> Rcpp::attributes::SourceFileAttributesParser::parseArguments (  
    const std::string & argText ) [private]
```

### 6.734.3.13 parseAttribute()

```
Attribute Rcpp::attributes::SourceFileAttributesParser::parseAttribute (  
    const std::vector< std::string > & match,  
    int lineNumber ) [private]
```

### 6.734.3.14 parseFunction()

```
Function Rcpp::attributes::SourceFileAttributesParser::parseFunction (  
    size_t lineNumber ) [private]
```

**6.734.3.15 parseParameters()**

```
std::vector<Param> Rcpp::attributes::SourceFileAttributesParser::parseParameters (
    const std::string & input ) [private]
```

**6.734.3.16 parseSignature()**

```
std::string Rcpp::attributes::SourceFileAttributesParser::parseSignature (
    size_t lineNumber ) [private]
```

**6.734.3.17 parseType()**

```
Type Rcpp::attributes::SourceFileAttributesParser::parseType (
    const std::string & text ) [private]
```

**6.734.3.18 rcppExportInvalidParameterWarning()**

```
void Rcpp::attributes::SourceFileAttributesParser::rcppExportInvalidParameterWarning (
    const std::string & param,
    size_t lineNumber ) [private]
```

**6.734.3.19 rcppExportNoFunctionFoundWarning()**

```
void Rcpp::attributes::SourceFileAttributesParser::rcppExportNoFunctionFoundWarning (
    size_t lineNumber ) [private]
```

**6.734.3.20 rcppExportWarning()**

```
void Rcpp::attributes::SourceFileAttributesParser::rcppExportWarning (
    const std::string & message,
    size_t lineNumber ) [private]
```

### 6.734.3.21 rcppInterfacesWarning()

```
void Rcpp::attributes::SourceFileAttributesParser::rcppInterfacesWarning (
    const std::string & message,
    size_t lineNumber ) [private]
```

### 6.734.3.22 roxygenChunks()

```
virtual const std::vector<std::vector<std::string> >& Rcpp::attributes::SourceFileAttributesParser::roxygenChunks ( ) const [inline], [virtual]
```

Implements [Rcpp::attributes::SourceFileAttributes](#).

Definition at line 510 of file attributes.cpp.

References [roxygenChunks\\_](#).

### 6.734.3.23 sourceDependencies()

```
const std::vector<FileInfo>& Rcpp::attributes::SourceFileAttributesParser::sourceDependencies ( )
const [inline]
```

Definition at line 547 of file attributes.cpp.

References [sourceDependencies\\_](#).

### 6.734.3.24 sourceFile()

```
virtual const std::string& Rcpp::attributes::SourceFileAttributesParser::sourceFile ( ) const
[inline], [virtual]
```

Implements [Rcpp::attributes::SourceFileAttributes](#).

Definition at line 499 of file attributes.cpp.

References [sourceFile\\_](#).

## 6.734.4 Member Data Documentation

#### 6.734.4.1 attributes\_

```
std::vector<Attribute> Rcpp::attributes::SourceFileAttributesParser::attributes_ [private]
```

Definition at line 578 of file attributes.cpp.

Referenced by begin(), end(), and hasGeneratorOutput().

#### 6.734.4.2 embeddedR\_

```
std::vector<std::string> Rcpp::attributes::SourceFileAttributesParser::embeddedR_ [private]
```

Definition at line 581 of file attributes.cpp.

Referenced by embeddedR().

#### 6.734.4.3 hasPackageInit\_

```
bool Rcpp::attributes::SourceFileAttributesParser::hasPackageInit_ [private]
```

Definition at line 580 of file attributes.cpp.

Referenced by hasPackageInit().

#### 6.734.4.4 lines\_

```
CharacterVector Rcpp::attributes::SourceFileAttributesParser::lines_ [private]
```

Definition at line 577 of file attributes.cpp.

#### 6.734.4.5 modules\_

```
std::vector<std::string> Rcpp::attributes::SourceFileAttributesParser::modules_ [private]
```

Definition at line 579 of file attributes.cpp.

Referenced by hasGeneratorOutput(), and modules().



#### 6.734.4.6 roxygenBuffer\_

```
std::vector<std::string> Rcpp::attributes::SourceFileAttributesParser::roxygenBuffer_ [private]
```

Definition at line 584 of file attributes.cpp.

#### 6.734.4.7 roxygenChunks\_

```
std::vector<std::vector<std::string> > Rcpp::attributes::SourceFileAttributesParser::roxygen↵  
Chunks_ [private]
```

Definition at line 583 of file attributes.cpp.

Referenced by hasGeneratorOutput(), and roxygenChunks().

#### 6.734.4.8 sourceDependencies\_

```
std::vector<FileInfo> Rcpp::attributes::SourceFileAttributesParser::sourceDependencies_ [private]
```

Definition at line 582 of file attributes.cpp.

Referenced by sourceDependencies().

#### 6.734.4.9 sourceFile\_

```
std::string Rcpp::attributes::SourceFileAttributesParser::sourceFile_ [private]
```

Definition at line 576 of file attributes.cpp.

Referenced by sourceFile().

The documentation for this class was generated from the following file:

- [src/attributes.cpp](#)

## 6.735 Rcpp::algorithm::helpers::sqrt Struct Reference

```
#include <algorithm.h>
```

## Public Member Functions

- `template<typename T >`  
`double operator() (T val)`

### 6.735.1 Detailed Description

Definition at line 217 of file `algorithm.h`.

### 6.735.2 Member Function Documentation

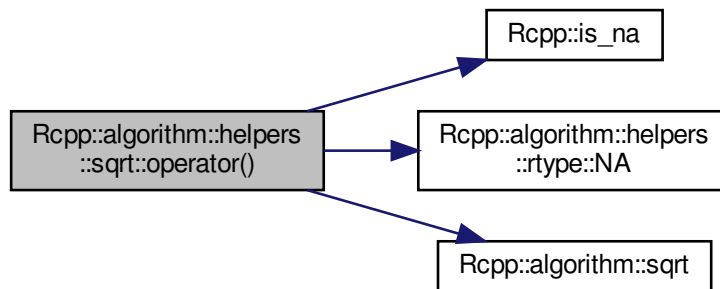
#### 6.735.2.1 operator>()

```
template<typename T >
double Rcpp::algorithm::helpers::sqrt::operator() (
    T val ) [inline]
```

Definition at line 219 of file `algorithm.h`.

References `Rcpp::is_na()`, `Rcpp::algorithm::helpers::rtype< T >::NA()`, and `Rcpp::algorithm::sqrt()`.

Here is the call graph for this function:

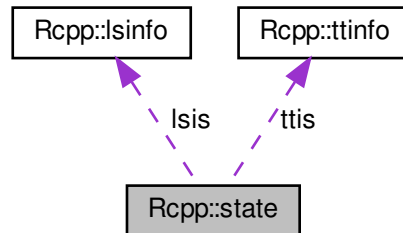


The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/algorithm.h`

## 6.736 Rcpp::state Struct Reference

Collaboration diagram for Rcpp::state:



### Public Attributes

- int [leapcnt](#)
- int [timecnt](#)
- int [typecnt](#)
- int [charcnt](#)
- int [goback](#)
- int [goahead](#)
- time\_t [ats](#) [TZ\_MAX\_TIMES]
- unsigned char [types](#) [TZ\_MAX\_TIMES]
- struct [ttinfo](#) [ttis](#) [TZ\_MAX\_TYPES]
- char [chars](#) [BIGGEST(BIGGEST(TZ\_MAX\_CHARS+1, sizeof [gmt](#)),(2 \*(MY\_TZNAME\_MAX+1)))]
- struct [lsinfo](#) [lsis](#) [TZ\_MAX\_LEAPS]

### 6.736.1 Detailed Description

Definition at line 393 of file `date.cpp`.

### 6.736.2 Member Data Documentation

#### 6.736.2.1 `ats`

```
time_t Rcpp::state::ats [TZ_MAX_TIMES]
```

Definition at line 400 of file `date.cpp`.

Referenced by `Rcpp::tzload()`, and `Rcpp::tzparse()`.

### 6.736.2.2 charcnt

```
int Rcpp::state::charcnt
```

Definition at line 397 of file date.cpp.

Referenced by Rcpp::tzload(), and Rcpp::tzparse().

### 6.736.2.3 chars

```
char Rcpp::state::chars[BIGGEST(BIGGEST(TZ_MAX_CHARS+1, sizeof gmtime), (2*(MY_TZNAME_MAX+1)))]
```

Definition at line 403 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

### 6.736.2.4 goahead

```
int Rcpp::state::goahead
```

Definition at line 399 of file date.cpp.

Referenced by Rcpp::tzload().

### 6.736.2.5 goback

```
int Rcpp::state::goback
```

Definition at line 398 of file date.cpp.

Referenced by Rcpp::tzload().

### 6.736.2.6 leapcnt

```
int Rcpp::state::leapcnt
```

Definition at line 394 of file date.cpp.

Referenced by Rcpp::timesub(), Rcpp::tzload(), and Rcpp::tzparse().

### 6.736.2.7 lsis

```
struct lsinfo Rcpp::state::lsis[TZ_MAX_LEAPS]
```

Definition at line 403 of file date.cpp.

Referenced by Rcpp::timesub(), and Rcpp::tzload().

### 6.736.2.8 timecnt

```
int Rcpp::state::timecnt
```

Definition at line 395 of file date.cpp.

Referenced by Rcpp::tzload(), and Rcpp::tzparse().

### 6.736.2.9 ttis

```
struct ttinfo Rcpp::state::ttis[TZ_MAX_TYPES]
```

Definition at line 401 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

### 6.736.2.10 typecnt

```
int Rcpp::state::typecnt
```

Definition at line 396 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

### 6.736.2.11 types

```
unsigned char Rcpp::state::types[TZ_MAX_TIMES]
```

Definition at line 401 of file date.cpp.

Referenced by Rcpp::tzload(), and Rcpp::tzparse().

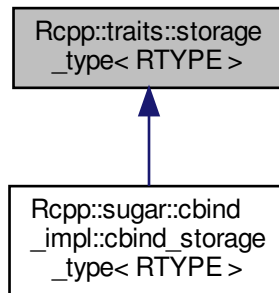
The documentation for this struct was generated from the following file:

- [src/date.cpp](#)

## 6.737 Rcpp::traits::storage\_type< RTYPE > Struct Template Reference

```
#include <storage_type.h>
```

Inheritance diagram for Rcpp::traits::storage\_type< RTYPE >:



### Public Types

- typedef SEXP [type](#)

### 6.737.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::traits::storage_type< RTYPE >
```

Indicates the storage type associated with a SEXP type for example: [storage\\_type<INTSXP>::type](#) is a typedef to int

The default is SEXP, which works for VECSXP, EXPRSXP and STRSXP

Definition at line 35 of file storage\_type.h.

### 6.737.2 Member Typedef Documentation

### 6.737.2.1 type

```
template<int RTYPE>
typedef SEXP Rcpp::traits::storage_type< RTYPE >::type
```

Definition at line 36 of file storage\_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/storage\_type.h

## 6.738 Rcpp::traits::storage\_type< CPLXSXP > Struct Reference

```
#include <storage_type.h>
```

### Public Types

- typedef Rcomplex [type](#)

### 6.738.1 Detailed Description

Total specialization for numeric vectors (CPLXSXP) typedef to Rcomplex

Definition at line 59 of file storage\_type.h.

### 6.738.2 Member Typedef Documentation

#### 6.738.2.1 type

```
typedef Rcomplex Rcpp::traits::storage_type< CPLXSXP >::type
```

Definition at line 60 of file storage\_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/storage\_type.h

## 6.739 Rcpp::traits::storage\_type< INTSXP > Struct Reference

```
#include <storage_type.h>
```

## Public Types

- typedef int [type](#)

### 6.739.1 Detailed Description

Total specialization for integer vector (INTSXP) typedef to int

Definition at line 43 of file storage\_type.h.

### 6.739.2 Member Typedef Documentation

#### 6.739.2.1 type

```
typedef int Rcpp::traits::storage\_type< INTSXP >::type
```

Definition at line 44 of file storage\_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[storage\\_type.h](#)

## 6.740 Rcpp::traits::storage\_type< LGLSXP > Struct Reference

```
#include <storage_type.h>
```

## Public Types

- typedef int [type](#)

### 6.740.1 Detailed Description

Total specialization for logical vectors (LGLSXP) typedef to int

Definition at line 75 of file storage\_type.h.

### 6.740.2 Member Typedef Documentation



### 6.740.2.1 type

```
typedef int Rcpp::traits::storage_type< LGLXP >::type
```

Definition at line 76 of file storage\_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/storage\_type.h

## 6.741 Rcpp::traits::storage\_type< RAWXP > Struct Reference

```
#include <storage_type.h>
```

### Public Types

- typedef Rbyte [type](#)

### 6.741.1 Detailed Description

Total specialization for raw vectors (RAWXP) typedef to Rbyte

Definition at line 67 of file storage\_type.h.

### 6.741.2 Member Typedef Documentation

#### 6.741.2.1 type

```
typedef Rbyte Rcpp::traits::storage_type< RAWXP >::type
```

Definition at line 68 of file storage\_type.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/storage\_type.h

## 6.742 Rcpp::traits::storage\_type< REALXP > Struct Reference

```
#include <storage_type.h>
```

## Public Types

- typedef double [type](#)

### 6.742.1 Detailed Description

Total specialization for numeric vectors (REALSXP) typedef to double

Definition at line 51 of file `storage_type.h`.

### 6.742.2 Member Typedef Documentation

#### 6.742.2.1 type

```
typedef double Rcpp::traits::storage\_type< REALSXP >::type
```

Definition at line 52 of file `storage_type.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/storage_type.h`

## 6.743 Rcpp::String Class Reference

```
#include <String.h>
```

### Public Types

- typedef [internal::string\\_proxy](#)< STRSXP > [StringProxy](#)
- typedef [internal::const\\_string\\_proxy](#)< STRSXP > [const\\_StringProxy](#)

## Public Member Functions

- [String](#) ()
- [String](#) (const [String](#) &s)
- [String](#) (SEXP charsexp)
- [String](#) (const [StringProxy](#) &proxy)
- [String](#) (const [StringProxy](#) &proxy, cetype\_t enc)
- [String](#) (const [const\\_StringProxy](#) &proxy)
- [String](#) (const [const\\_StringProxy](#) &proxy, cetype\_t enc)
- [String](#) (const std::string &s, cetype\_t enc=CE\_UTF8)
- [String](#) (const std::wstring &s, cetype\_t enc=CE\_UTF8)
- [String](#) (const char \*s, cetype\_t enc=CE\_UTF8)
- [String](#) (const wchar\_t \*s, cetype\_t enc=CE\_UTF8)
- [String](#) (int x)
- [String](#) (double x)
- [String](#) (bool x)
- [String](#) (Rcomplex x)
- [String](#) (Rbyte x)
- [~String](#) ()
- [String](#) & [operator=](#) (int x)
- [String](#) & [operator=](#) (double x)
- [String](#) & [operator=](#) (Rbyte x)
- [String](#) & [operator=](#) (bool x)
- [String](#) & [operator=](#) (Rcomplex x)
- [String](#) & [operator=](#) (SEXP x)
- [String](#) & [operator=](#) (const [StringProxy](#) &proxy)
- [String](#) & [operator=](#) (const [String](#) &other)
- [String](#) & [operator=](#) (const std::string &s)
- [String](#) & [operator=](#) (const char \*s)
- [String](#) & [operator=](#) (const std::wstring &s)
- [String](#) & [operator=](#) (const wchar\_t \*s)
- [String](#) & [operator+=](#) (const std::string &s)
- [String](#) & [operator+=](#) (const char \*s)
- [String](#) & [operator+=](#) (const std::wstring &s)
- [String](#) & [operator+=](#) (const wchar\_t \*s)
- [String](#) & [operator+=](#) (const [String](#) &other)
- [String](#) & [operator+=](#) (const [StringProxy](#) &proxy)
- [String](#) & [operator+=](#) (const [const\\_StringProxy](#) &proxy)
- [String](#) & [operator+=](#) (SEXP x)
- [String](#) & [replace\\_first](#) (const char \*s, const char \*news)
- [String](#) & [replace\\_first](#) (const [Rcpp::String](#) &s, const char \*news)
- [String](#) & [replace\\_first](#) (const char \*s, const [Rcpp::String](#) &news)
- [String](#) & [replace\\_first](#) (const [Rcpp::String](#) &s, const [Rcpp::String](#) &news)
- [String](#) & [replace\\_last](#) (const char \*s, const char \*news)
- [String](#) & [replace\\_last](#) (const [Rcpp::String](#) &s, const char \*news)
- [String](#) & [replace\\_last](#) (const char \*s, const [Rcpp::String](#) &news)
- [String](#) & [replace\\_last](#) (const [Rcpp::String](#) &s, const [Rcpp::String](#) &news)
- [String](#) & [replace\\_all](#) (const char \*s, const char \*news)
- [template](#)<typename LHS , typename RHS >  
[String](#) & [replace\\_all](#) (const LHS &s, const RHS &news)
- [String](#) & [replace\\_all](#) (const [Rcpp::String](#) &s, const char \*news)

- [String](#) & [replace\\_all](#) (const char \*s, const [Rcpp::String](#) &news)
- [String](#) & [replace\\_all](#) (const [Rcpp::String](#) &s, const [Rcpp::String](#) &news)
- [String](#) & [push\\_back](#) (const char \*s)
- [String](#) & [push\\_back](#) (const std::string &s)
- [String](#) & [push\\_back](#) (const [Rcpp::String](#) &s)
- [String](#) & [push\\_front](#) (const char \*s)
- [String](#) & [push\\_front](#) (const std::string &s)
- [String](#) & [push\\_front](#) (const [Rcpp::String](#) &s)
- void [set\\_na](#) ()
- SEXP [get\\_sexp\\_impl](#) () const
- SEXP [get\\_sexp](#) () const
- SEXP [get\\_sexp](#) ()
- [operator std::string](#) () const
- [operator std::wstring](#) () const
- const char \* [get\\_cstring](#) () const
- cetype\_t [get\\_encoding](#) () const
- void [set\\_encoding](#) (cetype\_t encoding)
- bool [operator<](#) (const [Rcpp::String](#) &other) const
- bool [operator==](#) (const [Rcpp::String](#) &other) const
- bool [operator!=](#) (const [Rcpp::String](#) &other) const
- bool [operator==](#) (const [StringProxy](#) &other) const
- bool [operator!=](#) (const [StringProxy](#) &other) const
- bool [operator==](#) (const [const\\_StringProxy](#) &other) const
- bool [operator!=](#) (const [const\\_StringProxy](#) &other) const
- bool [operator>](#) (const [Rcpp::String](#) &other) const
- bool [operator==](#) (SEXP other) const
- bool [operator!=](#) (SEXP other) const

## Private Member Functions

- template<typename T >  
[String](#) & [assign\\_wide\\_string](#) (const T &s)
- template<typename T >  
[String](#) & [append\\_wide\\_string](#) (const T &s)
- bool [is\\_na](#) () const
- void [setBuffer](#) ()
- void [setData](#) ()
- template<typename T >  
void [append](#) (const T &s)

## Private Attributes

- SEXP [data](#)
- SEXP [token](#)
- std::string [buffer](#)
- bool [valid](#)
- bool [buffer\\_ready](#)
- cetype\_t [enc](#)

## 6.743.1 Detailed Description

A single string, i.e. an element of a character vector. This represents CHARSXP SEXP

Definition at line 49 of file String.h.

## 6.743.2 Member Typedef Documentation

### 6.743.2.1 const\_StringProxy

```
typedef internal::const_string_proxy<STRSXP> Rcpp::String::const_StringProxy
```

Definition at line 52 of file String.h.

### 6.743.2.2 StringProxy

```
typedef internal::string_proxy<STRSXP> Rcpp::String::StringProxy
```

Definition at line 51 of file String.h.

## 6.743.3 Constructor & Destructor Documentation

### 6.743.3.1 String() [1/16]

```
Rcpp::String::String ( ) [inline]
```

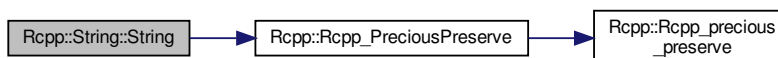
default constructor

Definition at line 55 of file String.h.

References data, Rcpp::Rcpp\_PreciousPreserve(), RCPP\_STRING\_DEBUG, and token.

Referenced by replace\_all().

Here is the call graph for this function:



**6.743.3.2 String()** [2/16]

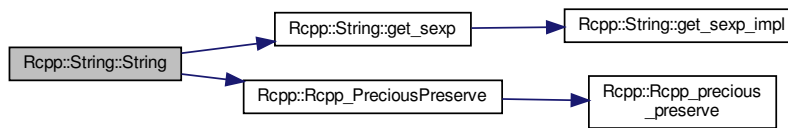
```
Rcpp::String::String (
    const String & s ) [inline]
```

copy constructor

Definition at line 61 of file String.h.

References `buffer_ready`, `data`, `get_sexp()`, `Rcpp::Rcpp_PreciousPreserve()`, `RCPP_STRING_DEBUG`, and `token`.

Here is the call graph for this function:

**6.743.3.3 String()** [3/16]

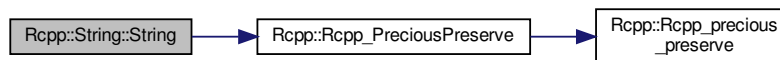
```
Rcpp::String::String (
    SEXP charsxp ) [inline]
```

construct a string from a single CHARXP SEXP

Definition at line 70 of file String.h.

References `buffer_ready`, `data`, `enc`, `Rcpp::Rcpp_PreciousPreserve()`, `RCPP_STRING_DEBUG`, `token`, and `valid`.

Here is the call graph for this function:



## 6.743.3.4 String() [4/16]

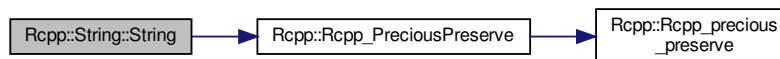
```
Rcpp::String::String (
    const StringProxy & proxy ) [inline]
```

from string proxy

Definition at line 93 of file String.h.

References data, Rcpp::Rcpp\_PreciousPreserve(), RCPP\_STRING\_DEBUG, and token.

Here is the call graph for this function:



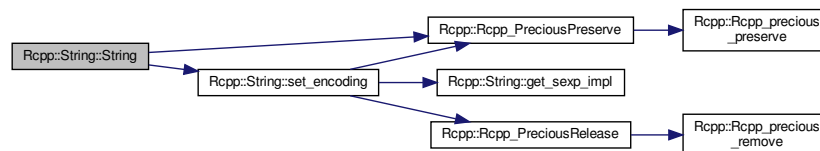
## 6.743.3.5 String() [5/16]

```
Rcpp::String::String (
    const StringProxy & proxy,
    cetype_t enc ) [inline]
```

Definition at line 98 of file String.h.

References data, enc, Rcpp::Rcpp\_PreciousPreserve(), RCPP\_STRING\_DEBUG, set\_encoding(), and token.

Here is the call graph for this function:



**6.743.3.6 String()** [6/16]

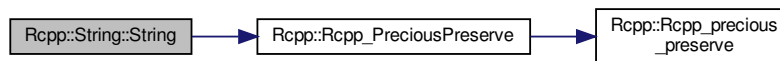
```
Rcpp::String::String (
    const const\_StringProxy & proxy ) [inline]
```

from string proxy

Definition at line 105 of file String.h.

References [data](#), [Rcpp::Rcpp\\_PreciousPreserve\(\)](#), [RCPP\\_STRING\\_DEBUG](#), and [token](#).

Here is the call graph for this function:

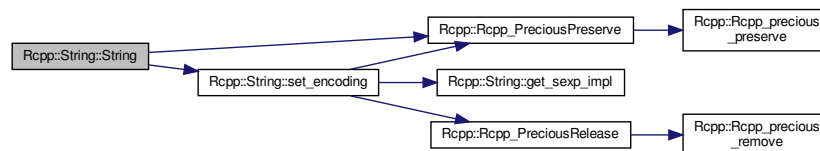
**6.743.3.7 String()** [7/16]

```
Rcpp::String::String (
    const const\_StringProxy & proxy,
    cetype_t enc ) [inline]
```

Definition at line 110 of file String.h.

References [data](#), [enc](#), [Rcpp::Rcpp\\_PreciousPreserve\(\)](#), [RCPP\\_STRING\\_DEBUG](#), [set\\_encoding\(\)](#), and [token](#).

Here is the call graph for this function:





### 6.743.3.8 String() [8/16]

```
Rcpp::String::String (  
    const std::string & s,  
    cetype_t enc = CE_UTF8 ) [inline]
```

from a std::string

Definition at line 117 of file String.h.

References RCPP\_STRING\_DEBUG.

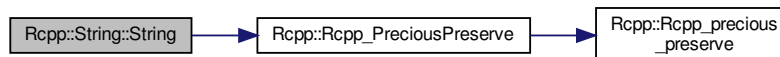
### 6.743.3.9 String() [9/16]

```
Rcpp::String::String (  
    const std::wstring & s,  
    cetype_t enc = CE_UTF8 ) [inline]
```

Definition at line 138 of file String.h.

References data, Rcpp::Rcpp\_PreciousPreserve(), RCPP\_STRING\_DEBUG, and token.

Here is the call graph for this function:



### 6.743.3.10 String() [10/16]

```
Rcpp::String::String (  
    const char * s,  
    cetype_t enc = CE_UTF8 ) [inline]
```

from a const char\*

Definition at line 144 of file String.h.

References data, RCPP\_STRING\_DEBUG, and token.

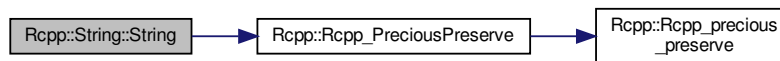
### 6.743.3.11 String() [11/16]

```
Rcpp::String::String (  
    const wchar_t * s,  
    cetype_t enc = CE_UTF8 ) [inline]
```

Definition at line 150 of file String.h.

References data, Rcpp::Rcpp\_PreciousPreserve(), RCPP\_STRING\_DEBUG, and token.

Here is the call graph for this function:



### 6.743.3.12 String() [12/16]

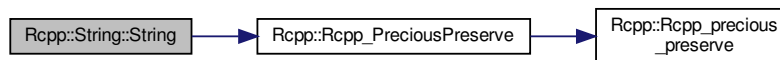
```
Rcpp::String::String (  
    int x ) [inline]
```

constructors from [R](#) primitives

Definition at line 156 of file String.h.

References data, Rcpp::Rcpp\_PreciousPreserve(), and token.

Here is the call graph for this function:



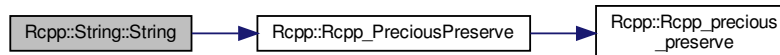
**6.743.3.13 String()** [13/16]

```
Rcpp::String::String (  
    double x ) [inline]
```

Definition at line 159 of file String.h.

References data, Rcpp::Rcpp\_PreciousPreserve(), and token.

Here is the call graph for this function:

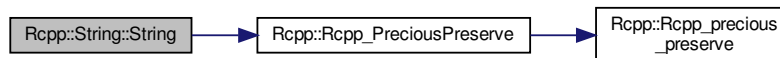
**6.743.3.14 String()** [14/16]

```
Rcpp::String::String (  
    bool x ) [inline]
```

Definition at line 162 of file String.h.

References data, Rcpp::Rcpp\_PreciousPreserve(), and token.

Here is the call graph for this function:



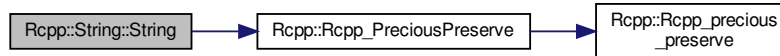
### 6.743.3.15 String() [15/16]

```
Rcpp::String::String (  
    Rcomplex x ) [inline]
```

Definition at line 165 of file String.h.

References data, Rcpp::Rcpp\_PreciousPreserve(), and token.

Here is the call graph for this function:



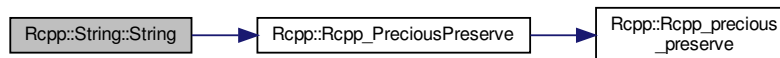
### 6.743.3.16 String() [16/16]

```
Rcpp::String::String (  
    Rbyte x ) [inline]
```

Definition at line 168 of file String.h.

References data, Rcpp::Rcpp\_PreciousPreserve(), and token.

Here is the call graph for this function:



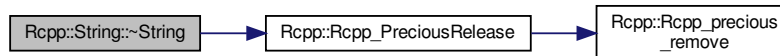
### 6.743.3.17 ~String()

```
Rcpp::String::~~String ( ) [inline]
```

Definition at line 172 of file String.h.

References data, Rcpp::Rcpp\_PreciousRelease(), and token.

Here is the call graph for this function:



## 6.743.4 Member Function Documentation

### 6.743.4.1 append()

```
template<typename T >
void Rcpp::String::append (
    const T & s ) [inline], [private]
```

Definition at line 672 of file String.h.

References buffer.

### 6.743.4.2 append\_wide\_string()

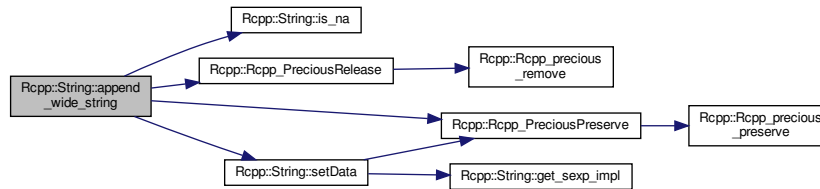
```
template<typename T >
String& Rcpp::String::append_wide_string (
    const T & s ) [inline], [private]
```

Definition at line 331 of file String.h.

References buffer\_ready, data, DEMANGLE, is\_na(), Rcpp::Rcpp\_PreciousPreserve(), Rcpp::Rcpp\_PreciousRelease(), RCPP\_STRING\_DEBUG\_1, setData(), token, and valid.

Referenced by operator+=().

Here is the call graph for this function:



#### 6.743.4.3 assign\_wide\_string()

```

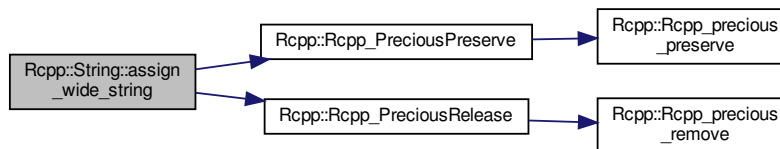
template<typename T >
String& Rcpp::String::assign_wide_string (
    const T & s ) [inline], [private]
  
```

Definition at line 303 of file String.h.

References `buffer_ready`, `data`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `token`, and `valid`.

Referenced by `operator=()`.

Here is the call graph for this function:



#### 6.743.4.4 get\_cstring()

```

const char* Rcpp::String::get_cstring ( ) const [inline]
  
```

Definition at line 576 of file String.h.

References `buffer`, `buffer_ready`, and `data`.

Referenced by `operator std::string()`, `operator std::wstring()`, `operator<()`, `operator>()`, `push_back()`, `push_front()`, `replace_all()`, `replace_first()`, `replace_last()`, and `Rcpp::trimws()`.

**6.743.4.5 get\_encoding()**

```
cetype_t Rcpp::String::get_encoding ( ) const [inline]
```

Definition at line 580 of file String.h.

References enc.

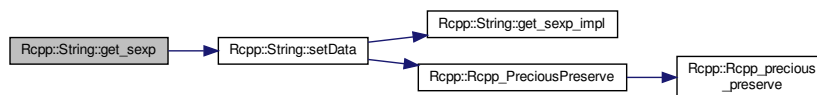
**6.743.4.6 get\_sexp() [1/2]**

```
SEXP Rcpp::String::get_sexp ( ) [inline]
```

Definition at line 562 of file String.h.

References data, RCPP\_STRING\_DEBUG\_1, setData(), and valid.

Here is the call graph for this function:

**6.743.4.7 get\_sexp() [2/2]**

```
SEXP Rcpp::String::get_sexp ( ) const [inline]
```

Definition at line 557 of file String.h.

References data, `get_sexp_impl()`, RCPP\_STRING\_DEBUG\_1, and valid.

Referenced by `Rcpp::internal::string_element_converter< RTYPE >::get()`, `Rcpp::internal::make_charsexp< Rcpp::↔String >()`, `operator!=()`, `operator=()`, `Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator=()`, `operator==()`, `Rcpp::internal::range_wrap_dispatch__impl__pair()`, `String()`, and `Rcpp::trimws()`.

Here is the call graph for this function:



#### 6.743.4.8 get\_sexp\_impl()

```
SEXPRcpp::String::get_sexp_impl ( ) const [inline]
```

Definition at line 544 of file String.h.

References buffer, and enc.

Referenced by get\_sexp(), set\_encoding(), and setData().

#### 6.743.4.9 is\_na()

```
bool Rcpp::String::is_na ( ) const [inline], [private]
```

Definition at line 657 of file String.h.

References data.

Referenced by append\_wide\_string(), operator+=(), push\_back(), push\_front(), replace\_all(), replace\_first(), and replace\_last().

#### 6.743.4.10 operator std::string()

```
Rcpp::String::operator std::string ( ) const [inline]
```

Definition at line 567 of file String.h.

References get\_cstring().

Here is the call graph for this function:





#### 6.743.4.11 operator std::wstring()

```
Rcpp::String::operator std::wstring ( ) const [inline]
```

Definition at line 571 of file String.h.

References `get_cstring()`.

Here is the call graph for this function:



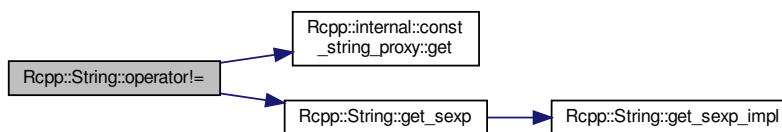
#### 6.743.4.12 operator"!="() [1/4]

```
bool Rcpp::String::operator!= (   
    const const_StringProxy & other ) const [inline]
```

Definition at line 623 of file String.h.

References `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::get()`, and `get_sexp()`.

Here is the call graph for this function:



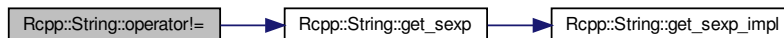
**6.743.4.13 operator"!=() [2/4]**

```
bool Rcpp::String::operator!= (
    const Rcpp::String & other ) const [inline]
```

Definition at line 607 of file String.h.

References `get_sexp()`.

Here is the call graph for this function:

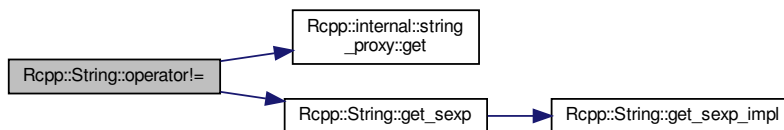
**6.743.4.14 operator"!=() [3/4]**

```
bool Rcpp::String::operator!= (
    const StringProxy & other ) const [inline]
```

Definition at line 615 of file String.h.

References `Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get()`, and `get_sexp()`.

Here is the call graph for this function:



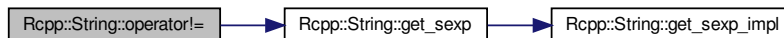
**6.743.4.15 operator"!="() [4/4]**

```
bool Rcpp::String::operator!=(  
    SEXP other ) const [inline]
```

Definition at line 635 of file String.h.

References `get_sexp()`.

Here is the call graph for this function:

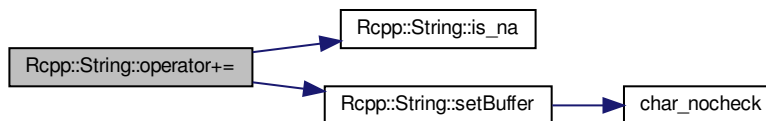
**6.743.4.16 operator+="() [1/8]**

```
String& Rcpp::String::operator+=(  
    const char * s ) [inline]
```

Definition at line 323 of file String.h.

References `buffer`, `is_na()`, `RCPP_STRING_DEBUG`, `setBuffer()`, and `valid`.

Here is the call graph for this function:



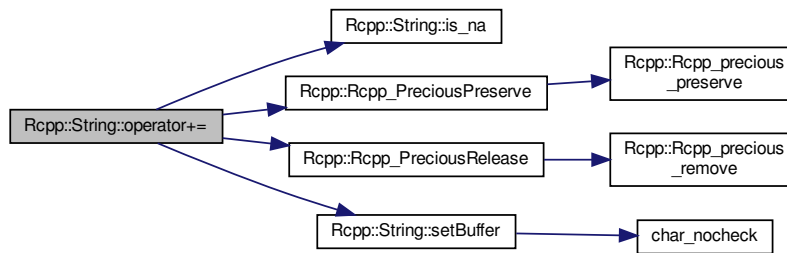
**6.743.4.17 operator+=() [2/8]**

```
String& Rcpp::String::operator+= (
    const const_StringProxy & proxy ) [inline]
```

Definition at line 380 of file String.h.

References `buffer`, `buffer_ready`, `data`, `is_na()`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `Rcpp_STRING_DEBUG`, `setBuffer()`, `token`, and `valid`.

Here is the call graph for this function:

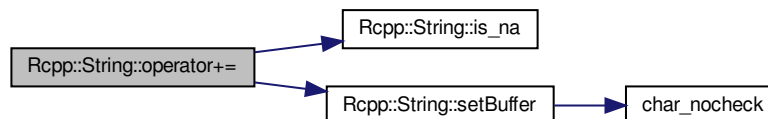
**6.743.4.18 operator+=() [3/8]**

```
String& Rcpp::String::operator+= (
    const std::string & s ) [inline]
```

Definition at line 316 of file String.h.

References `buffer`, `is_na()`, `Rcpp_STRING_DEBUG`, `setBuffer()`, and `valid`.

Here is the call graph for this function:



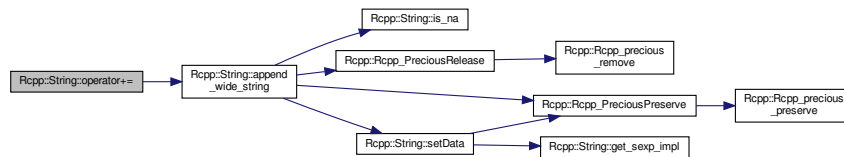
## 6.743.4.19 operator+=( ) [4/8]

```
String& Rcpp::String::operator+=(
    const std::wstring & s ) [inline]
```

Definition at line 348 of file String.h.

References `append_wide_string()`.

Here is the call graph for this function:



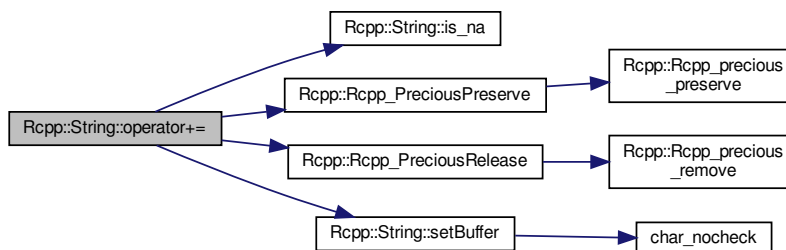
## 6.743.4.20 operator+=( ) [5/8]

```
String& Rcpp::String::operator+=(
    const String & other ) [inline]
```

Definition at line 351 of file String.h.

References `buffer`, `buffer_ready`, `data`, `is_na()`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `Rcpp::Rcpp_PreciousRemove()`, `Rcpp::Rcpp_PreciousPreserve`, `Rcpp::Rcpp_PreciousRemove`, `RCPP_STRING_DEBUG`, `setBuffer()`, `token`, and `valid`.

Here is the call graph for this function:



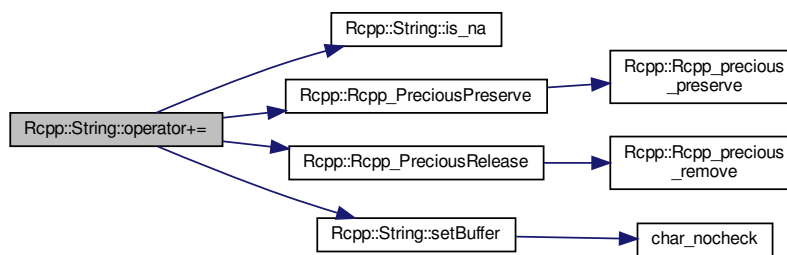
**6.743.4.21 operator+=()** [6/8]

```
String& Rcpp::String::operator+= (
    const StringProxy & proxy ) [inline]
```

Definition at line 365 of file String.h.

References `buffer`, `buffer_ready`, `data`, `is_na()`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `RCPPE_STRING_DEBUG`, `setBuffer()`, `token`, and `valid`.

Here is the call graph for this function:

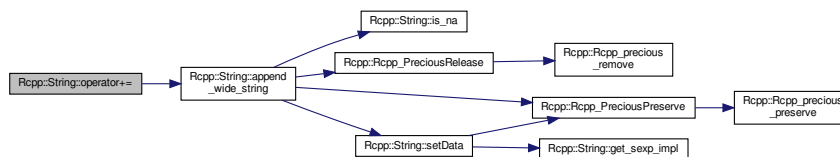
**6.743.4.22 operator+=()** [7/8]

```
String& Rcpp::String::operator+= (
    const wchar_t * s ) [inline]
```

Definition at line 349 of file String.h.

References `append_wide_string()`.

Here is the call graph for this function:



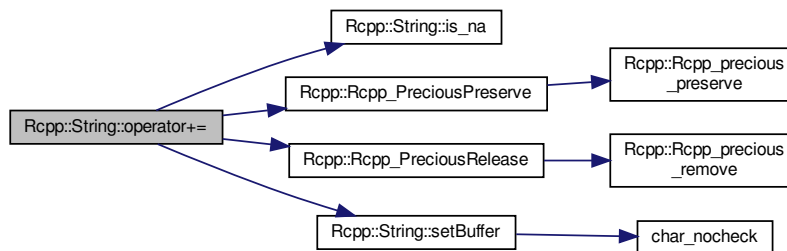
**6.743.4.23 operator+=( ) [8/8]**

```
String& Rcpp::String::operator+= (
    SEXP x ) [inline]
```

Definition at line 395 of file String.h.

References `buffer`, `buffer_ready`, `data`, `is_na()`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `Rcpp_STRING_DEBUG`, `setBuffer()`, `token`, and `valid`.

Here is the call graph for this function:

**6.743.4.24 operator<()**

```
bool Rcpp::String::operator< (
    const Rcpp::String & other ) const [inline]
```

Definition at line 600 of file String.h.

References `get_cstring()`.

Here is the call graph for this function:



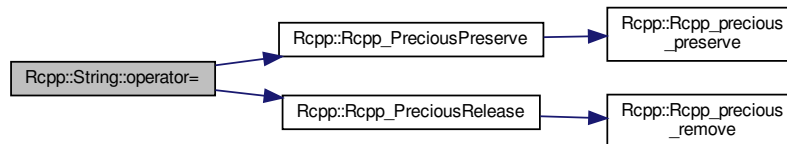
**6.743.4.25 operator=()** [1/12]

```
String& Rcpp::String::operator= (
    bool x ) [inline]
```

Definition at line 203 of file String.h.

References `buffer_ready`, `data`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `token`, and `valid`.

Here is the call graph for this function:

**6.743.4.26 operator=()** [2/12]

```
String& Rcpp::String::operator= (
    const char * s ) [inline]
```

Definition at line 294 of file String.h.

References `buffer`, `buffer_ready`, and `valid`.

**6.743.4.27 operator=()** [3/12]

```
String& Rcpp::String::operator= (
    const std::string & s ) [inline]
```

Definition at line 264 of file String.h.

References `buffer`, `buffer_ready`, and `valid`.



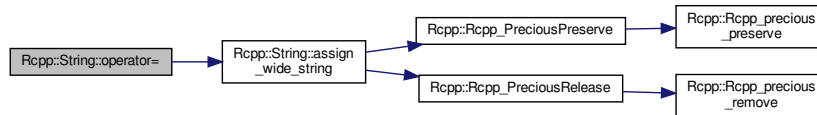
## 6.743.4.28 operator=() [4/12]

```
String& Rcpp::String::operator= (
    const std::wstring & s ) [inline]
```

Definition at line 313 of file String.h.

References `assign_wide_string()`.

Here is the call graph for this function:



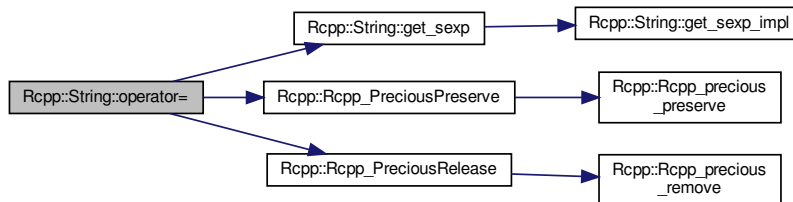
## 6.743.4.29 operator=() [5/12]

```
String& Rcpp::String::operator= (
    const String & other ) [inline]
```

Definition at line 240 of file String.h.

References `buffer`, `buffer_ready`, `data`, `enc`, `get_sexp()`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `token`, and `valid`.

Here is the call graph for this function:



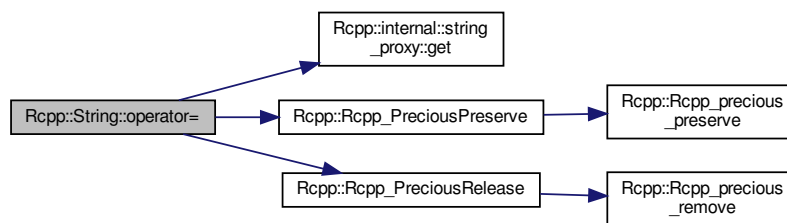
**6.743.4.30 operator=()** [6/12]

```
String& Rcpp::String::operator= (
    const StringProxy & proxy ) [inline]
```

Definition at line 229 of file String.h.

References `buffer_ready`, `data`, `Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get()`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `token`, and `valid`.

Here is the call graph for this function:

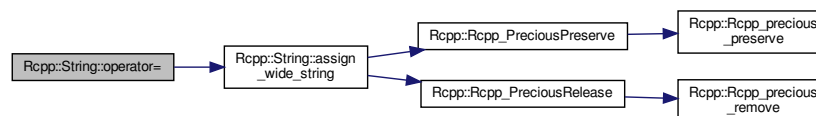
**6.743.4.31 operator=()** [7/12]

```
String& Rcpp::String::operator= (
    const wchar_t * s ) [inline]
```

Definition at line 314 of file String.h.

References `assign_wide_string()`.

Here is the call graph for this function:



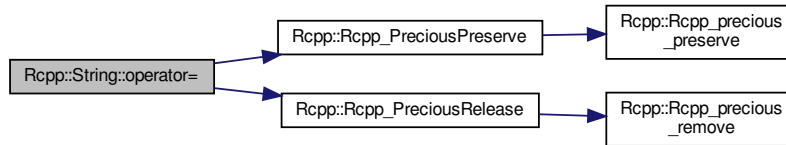
**6.743.4.32 operator=()** [8/12]

```
String& Rcpp::String::operator= (  
    double x ) [inline]
```

Definition at line 187 of file String.h.

References `buffer_ready`, `data`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `token`, and `valid`.

Here is the call graph for this function:

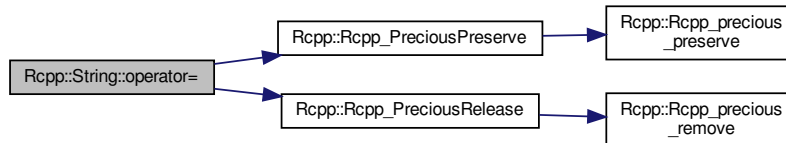
**6.743.4.33 operator=()** [9/12]

```
String& Rcpp::String::operator= (  
    int x ) [inline]
```

Definition at line 179 of file String.h.

References `buffer_ready`, `data`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `token`, and `valid`.

Here is the call graph for this function:



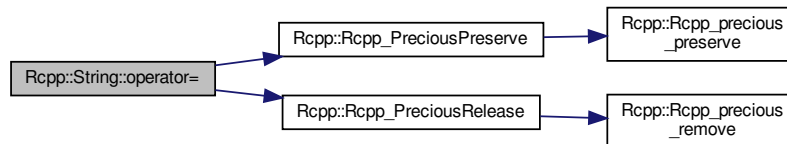
**6.743.4.34 operator=()** [10/12]

```
String& Rcpp::String::operator= (
    Rbyte x ) [inline]
```

Definition at line 195 of file String.h.

References `buffer_ready`, `data`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `token`, and `valid`.

Here is the call graph for this function:

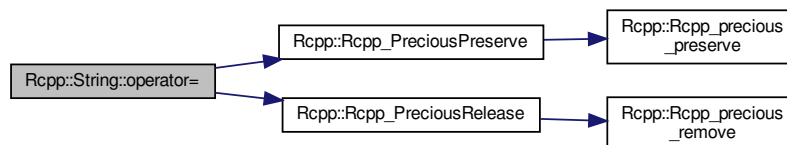
**6.743.4.35 operator=()** [11/12]

```
String& Rcpp::String::operator= (
    Rcomplex x ) [inline]
```

Definition at line 211 of file String.h.

References `buffer_ready`, `data`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `token`, and `valid`.

Here is the call graph for this function:



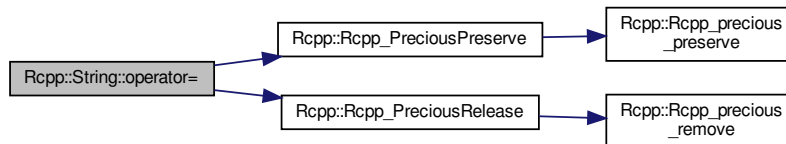
**6.743.4.36 operator=()** [12/12]

```
String& Rcpp::String::operator= (
    SEXP x ) [inline]
```

Definition at line 219 of file String.h.

References `buffer_ready`, `data`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `token`, and `valid`.

Here is the call graph for this function:

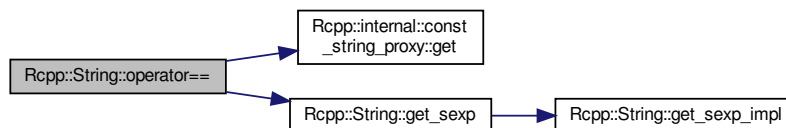
**6.743.4.37 operator==()** [1/4]

```
bool Rcpp::String::operator== (
    const const_StringProxy & other ) const [inline]
```

Definition at line 619 of file String.h.

References `Rcpp::internal::const_string_proxy< RTYPE, StoragePolicy >::get()`, and `get_sexp()`.

Here is the call graph for this function:



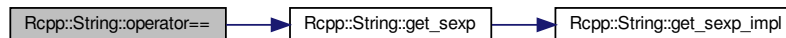
**6.743.4.38 operator==( ) [2/4]**

```
bool Rcpp::String::operator==(
    const Rcpp::String & other ) const [inline]
```

Definition at line 604 of file String.h.

References `get_sexp()`.

Here is the call graph for this function:

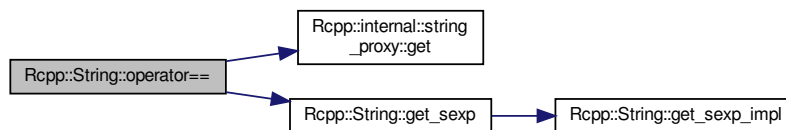
**6.743.4.39 operator==( ) [3/4]**

```
bool Rcpp::String::operator==(
    const StringProxy & other ) const [inline]
```

Definition at line 611 of file String.h.

References `Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get()`, and `get_sexp()`.

Here is the call graph for this function:



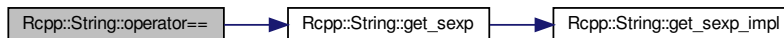
**6.743.4.40 operator==( ) [4/4]**

```
bool Rcpp::String::operator==(
    SEXP other ) const [inline]
```

Definition at line 631 of file String.h.

References `get_sexp()`.

Here is the call graph for this function:

**6.743.4.41 operator>( )**

```
bool Rcpp::String::operator>(
    const Rcpp::String & other ) const [inline]
```

Definition at line 627 of file String.h.

References `get_cstring()`.

Here is the call graph for this function:



**6.743.4.42 push\_back()** [1/3]

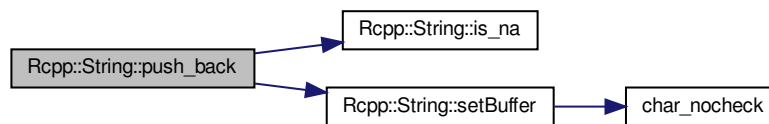
```
String& Rcpp::String::push_back (
    const char * s ) [inline]
```

Definition at line 506 of file String.h.

References `buffer`, `is_na()`, `setBuffer()`, and `valid`.

Referenced by `push_back()`.

Here is the call graph for this function:

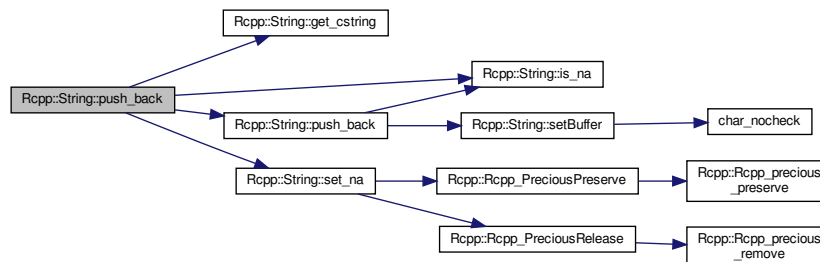
**6.743.4.43 push\_back()** [2/3]

```
String& Rcpp::String::push_back (
    const Rcpp::String & s ) [inline]
```

Definition at line 514 of file String.h.

References `get_cstring()`, `is_na()`, `push_back()`, and `set_na()`.

Here is the call graph for this function:





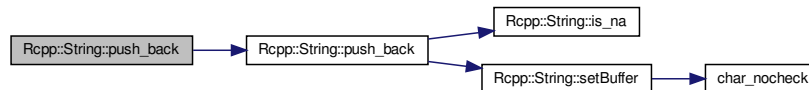
#### 6.743.4.44 push\_back() [3/3]

```
String& Rcpp::String::push_back (
    const std::string & s ) [inline]
```

Definition at line 511 of file String.h.

References push\_back().

Here is the call graph for this function:



#### 6.743.4.45 push\_front() [1/3]

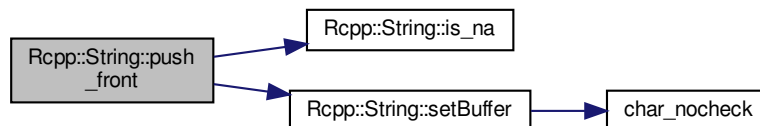
```
String& Rcpp::String::push_front (
    const char * s ) [inline]
```

Definition at line 520 of file String.h.

References buffer, is\_na(), setBuffer(), and valid.

Referenced by push\_front().

Here is the call graph for this function:



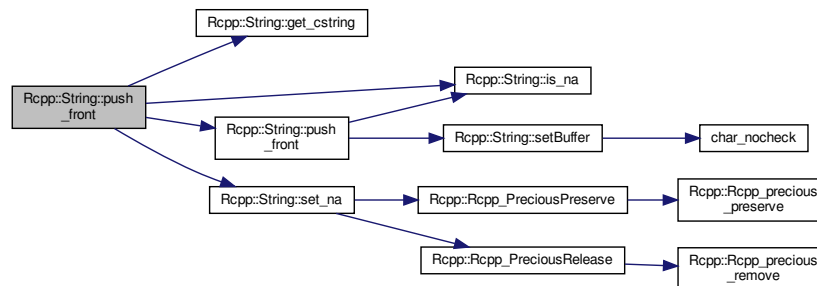
**6.743.4.46 push\_front()** [2/3]

```
String& Rcpp::String::push_front (
    const Rcpp::String & s ) [inline]
```

Definition at line 528 of file String.h.

References `get_cstring()`, `is_na()`, `push_front()`, and `set_na()`.

Here is the call graph for this function:

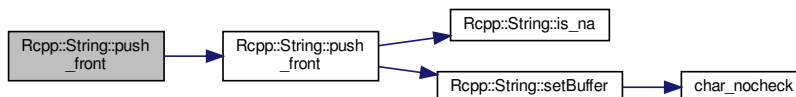
**6.743.4.47 push\_front()** [3/3]

```
String& Rcpp::String::push_front (
    const std::string & s ) [inline]
```

Definition at line 525 of file String.h.

References `push_front()`.

Here is the call graph for this function:



**6.743.4.48** `replace_all()` [1/5]

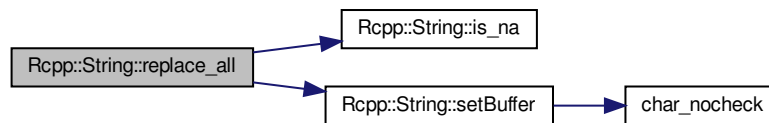
```
String& Rcpp::String::replace_all (
    const char * s,
    const char * news ) [inline]
```

Definition at line 469 of file String.h.

References `buffer`, `is_na()`, `RCPP_STRING_DEBUG_2`, `setBuffer()`, and `valid`.

Referenced by `replace_all()`.

Here is the call graph for this function:

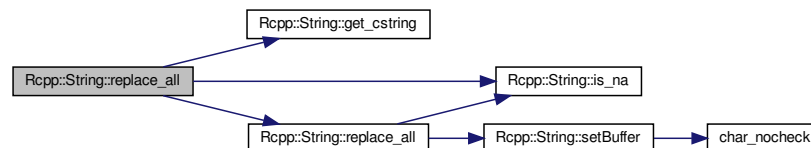
**6.743.4.49** `replace_all()` [2/5]

```
String& Rcpp::String::replace_all (
    const char * s,
    const Rcpp::String & news ) [inline]
```

Definition at line 495 of file String.h.

References `get_cstring()`, `is_na()`, and `replace_all()`.

Here is the call graph for this function:



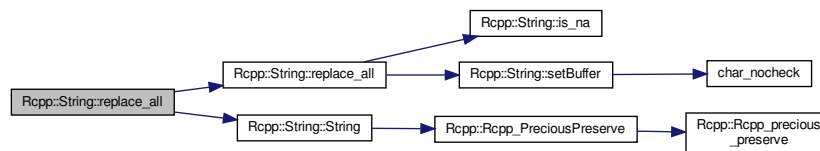
**6.743.4.50 replace\_all()** [3/5]

```
template<typename LHS , typename RHS >
String& Rcpp::String::replace_all (
    const LHS & s,
    const RHS & news ) [inline]
```

Definition at line 486 of file String.h.

References `replace_all()`, and `String()`.

Here is the call graph for this function:

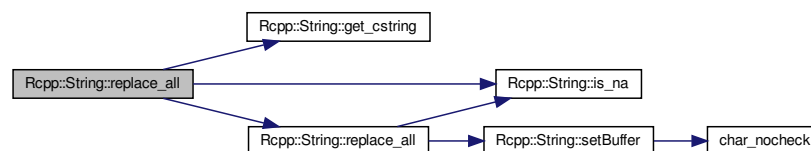
**6.743.4.51 replace\_all()** [4/5]

```
String& Rcpp::String::replace_all (
    const Rcpp::String & s,
    const char * news ) [inline]
```

Definition at line 490 of file String.h.

References `get_cstring()`, `is_na()`, and `replace_all()`.

Here is the call graph for this function:



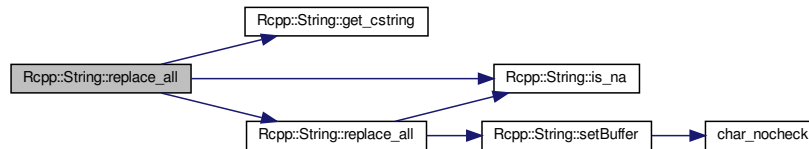
6.743.4.52 `replace_all()` [5/5]

```
String& Rcpp::String::replace_all (
    const Rcpp::String & s,
    const Rcpp::String & news ) [inline]
```

Definition at line 500 of file String.h.

References `get_cstring()`, `is_na()`, and `replace_all()`.

Here is the call graph for this function:

6.743.4.53 `replace_first()` [1/4]

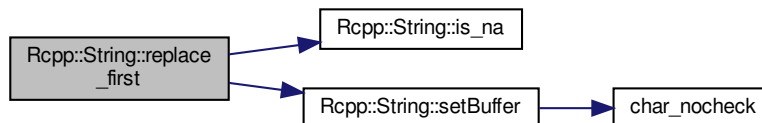
```
String& Rcpp::String::replace_first (
    const char * s,
    const char * news ) [inline]
```

Definition at line 416 of file String.h.

References `buffer`, `is_na()`, `RCPP_STRING_DEBUG_2`, `setBuffer()`, and `valid`.

Referenced by `replace_first()`.

Here is the call graph for this function:



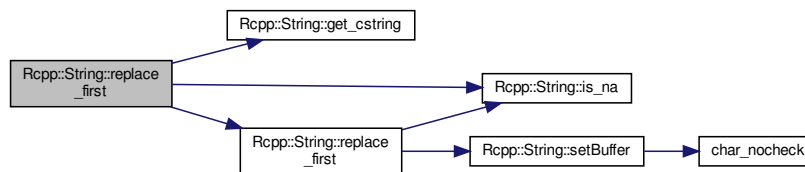
**6.743.4.54** `replace_first()` [2/4]

```
String& Rcpp::String::replace_first (
    const char * s,
    const Rcpp::String & news ) [inline]
```

Definition at line 431 of file String.h.

References `get_cstring()`, `is_na()`, and `replace_first()`.

Here is the call graph for this function:

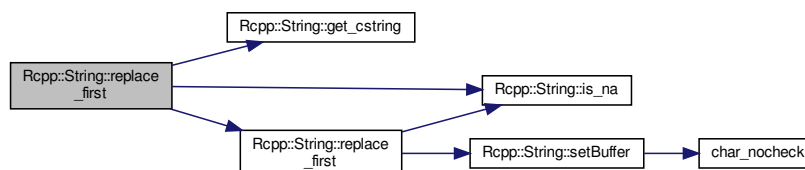
**6.743.4.55** `replace_first()` [3/4]

```
String& Rcpp::String::replace_first (
    const Rcpp::String & s,
    const char * news ) [inline]
```

Definition at line 426 of file String.h.

References `get_cstring()`, `is_na()`, and `replace_first()`.

Here is the call graph for this function:



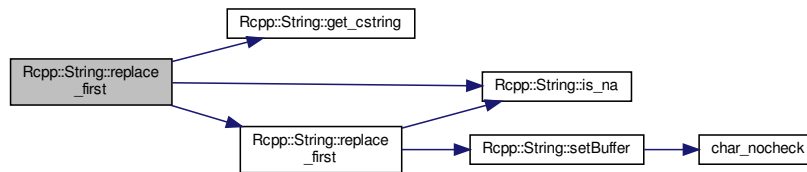
## 6.743.4.56 replace\_first() [4/4]

```
String& Rcpp::String::replace_first (
    const Rcpp::String & s,
    const Rcpp::String & news ) [inline]
```

Definition at line 436 of file String.h.

References `get_cstring()`, `is_na()`, and `replace_first()`.

Here is the call graph for this function:



## 6.743.4.57 replace\_last() [1/4]

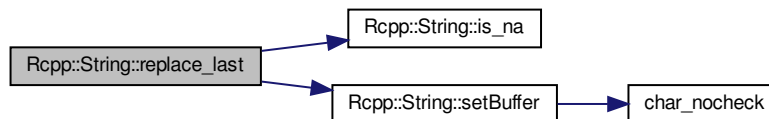
```
String& Rcpp::String::replace_last (
    const char * s,
    const char * news ) [inline]
```

Definition at line 442 of file String.h.

References `buffer`, `is_na()`, `RCPP_STRING_DEBUG_2`, `setBuffer()`, and `valid`.

Referenced by `replace_last()`.

Here is the call graph for this function:



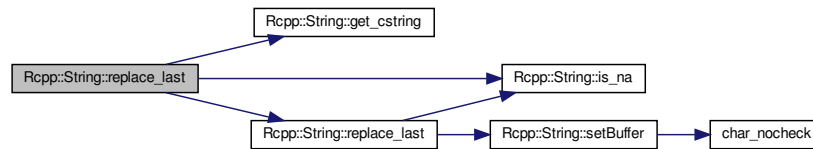
**6.743.4.58 replace\_last()** [2/4]

```
String& Rcpp::String::replace_last (
    const char * s,
    const Rcpp::String & news ) [inline]
```

Definition at line 457 of file String.h.

References `get_cstring()`, `is_na()`, and `replace_last()`.

Here is the call graph for this function:

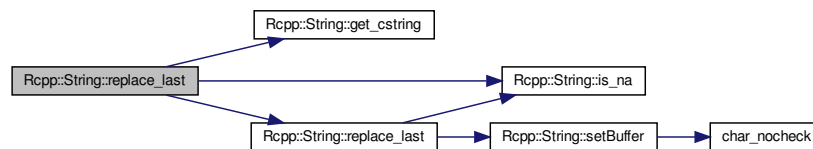
**6.743.4.59 replace\_last()** [3/4]

```
String& Rcpp::String::replace_last (
    const Rcpp::String & s,
    const char * news ) [inline]
```

Definition at line 452 of file String.h.

References `get_cstring()`, `is_na()`, and `replace_last()`.

Here is the call graph for this function:





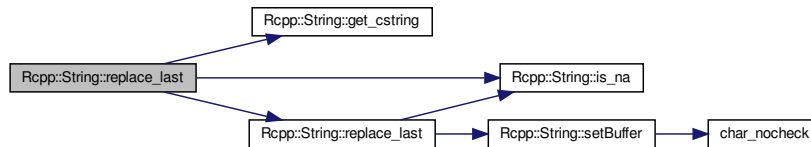
6.743.4.60 `replace_last()` [4/4]

```
String& Rcpp::String::replace_last (
    const Rcpp::String & s,
    const Rcpp::String & news ) [inline]
```

Definition at line 462 of file String.h.

References `get_cstring()`, `is_na()`, and `replace_last()`.

Here is the call graph for this function:

6.743.4.61 `set_encoding()`

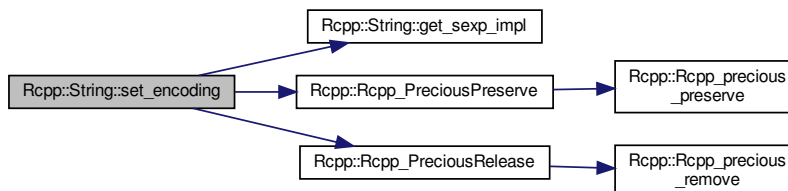
```
void Rcpp::String::set_encoding (
    cetype_t encoding ) [inline]
```

Definition at line 584 of file String.h.

References `data`, `enc`, `get_sexp_impl()`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `token`, and `valid`.

Referenced by `String()`.

Here is the call graph for this function:



**6.743.4.62 set\_na()**

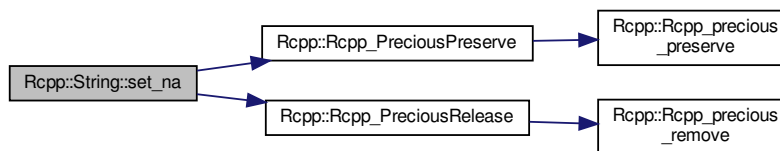
```
void Rcpp::String::set_na ( ) [inline]
```

Definition at line 535 of file String.h.

References `buffer_ready`, `data`, `Rcpp::Rcpp_PreciousPreserve()`, `Rcpp::Rcpp_PreciousRelease()`, `token`, and `valid`.

Referenced by `push_back()`, and `push_front()`.

Here is the call graph for this function:

**6.743.4.63 setBuffer()**

```
void Rcpp::String::setBuffer ( ) [inline], [private]
```

Definition at line 658 of file String.h.

References `buffer`, `buffer_ready`, `char_nocheck()`, and `data`.

Referenced by `operator+=()`, `push_back()`, `push_front()`, `replace_all()`, `replace_first()`, and `replace_last()`.

Here is the call graph for this function:



**6.743.4.64 setData()**

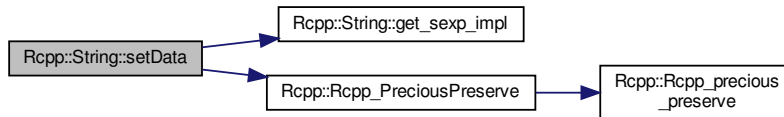
```
void Rcpp::String::setData ( ) [inline], [private]
```

Definition at line 664 of file String.h.

References `data`, `get_sexp_impl()`, `Rcpp::Rcpp_PreciousPreserve()`, `RCPP_STRING_DEBUG`, `token`, and `valid`.

Referenced by `append_wide_string()`, and `get_sexp()`.

Here is the call graph for this function:

**6.743.5 Member Data Documentation****6.743.5.1 buffer**

```
std::string Rcpp::String::buffer [private]
```

a buffer used to do string operations without going back to the SEXP

Definition at line 646 of file String.h.

Referenced by `append()`, `get_cstring()`, `get_sexp_impl()`, `operator+=()`, `operator=()`, `push_back()`, `push_front()`, `replace_all()`, `replace_first()`, `replace_last()`, and `setBuffer()`.

**6.743.5.2 buffer\_ready**

```
bool Rcpp::String::buffer_ready [private]
```

is the buffer initialized

Definition at line 652 of file String.h.

Referenced by `append_wide_string()`, `assign_wide_string()`, `get_cstring()`, `operator+=()`, `operator=()`, `set_na()`, `setBuffer()`, and `String()`.

### 6.743.5.3 data

```
SEXPRcpp::String::data [private]
```

the CHARSEXPR this [String](#) encapsulates

Definition at line 642 of file [String.h](#).

Referenced by [append\\_wide\\_string\(\)](#), [assign\\_wide\\_string\(\)](#), [get\\_cstring\(\)](#), [get\\_sexpr\(\)](#), [is\\_na\(\)](#), [operator+=\(\)](#), [operator=\(\)](#), [set\\_encoding\(\)](#), [set\\_na\(\)](#), [setBuffer\(\)](#), [setData\(\)](#), [String\(\)](#), and [~String\(\)](#).

### 6.743.5.4 enc

```
cetype_t Rcpp::String::enc [private]
```

the encoding of encapsulated CHARSEXPR

Definition at line 655 of file [String.h](#).

Referenced by [get\\_encoding\(\)](#), [get\\_sexpr\\_impl\(\)](#), [operator=\(\)](#), [set\\_encoding\(\)](#), and [String\(\)](#).

### 6.743.5.5 token

```
SEXPRcpp::String::token [private]
```

Definition at line 643 of file [String.h](#).

Referenced by [append\\_wide\\_string\(\)](#), [assign\\_wide\\_string\(\)](#), [operator+=\(\)](#), [operator=\(\)](#), [set\\_encoding\(\)](#), [set\\_na\(\)](#), [setData\(\)](#), [String\(\)](#), and [~String\(\)](#).

### 6.743.5.6 valid

```
bool Rcpp::String::valid [private]
```

is data in sync with buffer

Definition at line 649 of file [String.h](#).

Referenced by [append\\_wide\\_string\(\)](#), [assign\\_wide\\_string\(\)](#), [get\\_sexpr\(\)](#), [operator+=\(\)](#), [operator=\(\)](#), [push\\_back\(\)](#), [push\\_front\(\)](#), [replace\\_all\(\)](#), [replace\\_first\(\)](#), [replace\\_last\(\)](#), [set\\_encoding\(\)](#), [set\\_na\(\)](#), [setData\(\)](#), and [String\(\)](#).

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/String.h](#)

## 6.744 Rcpp::internal::string\_element\_converter< RTYPE > Class Template Reference

```
#include <converter.h>
```

### Public Types

- typedef SEXP [target](#)

### Static Public Member Functions

- `template<typename T >`  
static SEXP [get](#) (const T &input)
- static SEXP [get](#) (const std::string &input)
- static SEXP [get](#) (const [Rcpp::String](#) &input)
- static SEXP [get](#) (const char &input)
- static SEXP [get](#) (SEXP x)

### 6.744.1 Detailed Description

```
template<int RTYPE>  
class Rcpp::internal::string_element_converter< RTYPE >
```

Definition at line 43 of file converter.h.

### 6.744.2 Member Typedef Documentation

#### 6.744.2.1 target

```
template<int RTYPE>  
typedef SEXP Rcpp::internal::string\_element\_converter< RTYPE >::target
```

Definition at line 45 of file converter.h.

### 6.744.3 Member Function Documentation

**6.744.3.1 get() [1/5]**

```
template<int RTYPE>
static SEXP Rcpp::internal::string_element_converter< RTYPE >::get (
    const char & input ) [inline], [static]
```

Definition at line 61 of file converter.h.

References RCPP\_DEBUG.

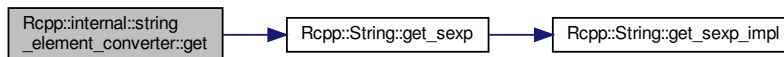
**6.744.3.2 get() [2/5]**

```
template<int RTYPE>
SEXP Rcpp::internal::string_element_converter< RTYPE >::get (
    const Rcpp::String & input ) [static]
```

Definition at line 688 of file String.h.

References Rcpp::String::get\_sexp(), and RCPP\_DEBUG.

Here is the call graph for this function:

**6.744.3.3 get() [3/5]**

```
template<int RTYPE>
static SEXP Rcpp::internal::string_element_converter< RTYPE >::get (
    const std::string & input ) [inline], [static]
```

Definition at line 54 of file converter.h.

References RCPP\_DEBUG.

**6.744.3.4** `get()` [4/5]

```
template<int RTYPE>
template<typename T >
static SEXP Rcpp::internal::string_element_converter< RTYPE >::get (
    const T & input ) [inline], [static]
```

Definition at line 48 of file converter.h.

References DEMANGLE, and RCPP\_DEBUG\_1.

**6.744.3.5** `get()` [5/5]

```
template<int RTYPE>
static SEXP Rcpp::internal::string_element_converter< RTYPE >::get (
    SEXP x ) [inline], [static]
```

Definition at line 67 of file converter.h.

References RCPP\_DEBUG.

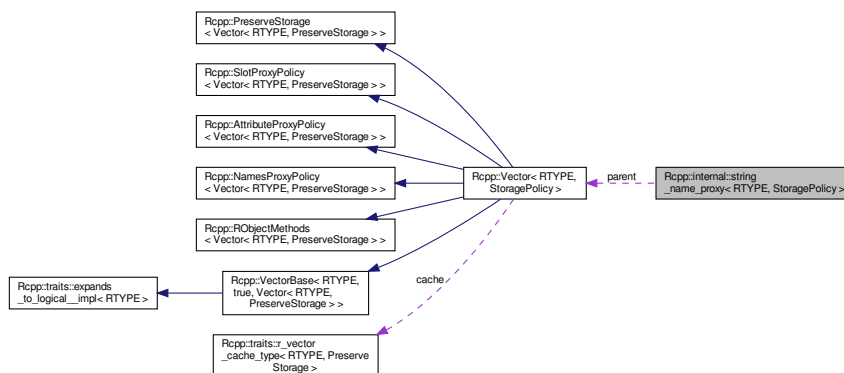
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/[converter.h](#)
- inst/include/Rcpp/[String.h](#)

**6.745 Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy > Class Template Reference**

```
#include <proxy.h>
```

Collaboration diagram for Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >:



## Public Types

- typedef `Rcpp::Vector`< RTYPE, StoragePolicy > `VECTOR`
- typedef const char \* `iterator`
- typedef const char & `reference`

## Public Member Functions

- `string_name_proxy` (`VECTOR` &v, const std::string &name\_)
- `string_name_proxy` (const `string_name_proxy` &other)
- `~string_name_proxy` ()
- `string_name_proxy` & `operator=` (const std::string &rhs)
- `string_name_proxy` & `operator=` (const `string_name_proxy` &other)
- `operator char *` () const
- `operator SEXP` () const
- `iterator begin` ()
- `iterator end` ()
- `reference operator[]` (R\_xlen\_t i)
- `R_xlen_t size` ()

## Private Member Functions

- void `set` (const std::string &rhs)
- char \* `get` () const

## Private Attributes

- `VECTOR` & `parent`
- std::string `name`

### 6.745.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>
class Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >
```

Definition at line 84 of file proxy.h.

### 6.745.2 Member Typedef Documentation



### 6.745.2.1 iterator

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const char* Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::iterator
```

Definition at line 87 of file proxy.h.

### 6.745.2.2 reference

```
template<int RTYPE, template< class > class StoragePolicy>
typedef const char& Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::reference
```

Definition at line 88 of file proxy.h.

### 6.745.2.3 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::string_name_proxy< RTYPE, Storage←
Policy >::VECTOR
```

Definition at line 86 of file proxy.h.

## 6.745.3 Constructor & Destructor Documentation

### 6.745.3.1 string\_name\_proxy() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::string_name_proxy (
    VECTOR & v,
    const std::string & name_ ) [inline]
```

Definition at line 90 of file proxy.h.

### 6.745.3.2 string\_name\_proxy() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::string_name_proxy (
    const string_name_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 92 of file proxy.h.

### 6.745.3.3 ~string\_name\_proxy()

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::~string_name_proxy ( ) [inline]
```

Definition at line 94 of file proxy.h.

## 6.745.4 Member Function Documentation

### 6.745.4.1 begin()

```
template<int RTYPE, template< class > class StoragePolicy>
iterator Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::begin ( ) [inline]
```

Definition at line 113 of file proxy.h.

References `Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::get()`.

Referenced by `Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::end()`.

Here is the call graph for this function:



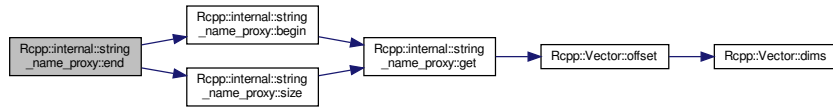
## 6.745.4.2 end()

```
template<int RTYPE, template< class > class StoragePolicy>
iterator Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::end ( ) [inline]
```

Definition at line 114 of file proxy.h.

References Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::begin(), and Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



## 6.745.4.3 get()

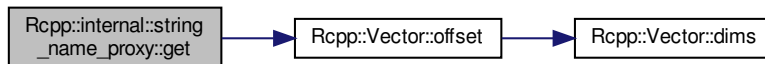
```
template<int RTYPE, template< class > class StoragePolicy>
char* Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::get ( ) const [inline], [private]
```

Definition at line 130 of file proxy.h.

References Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::name, Rcpp::Vector< RTYPE, StoragePolicy >::offset(), and Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::begin(), Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::operator char \*(), Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::operator SEXP(), Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::operator=(), Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::operator[](), and Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



#### 6.745.4.4 operator char \*()

```
template<int RTYPE, template< class > class StoragePolicy>  
Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator char * ( ) const [inline]
```

Definition at line 105 of file proxy.h.

References Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



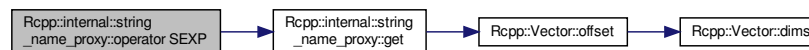
#### 6.745.4.5 operator SEXP()

```
template<int RTYPE, template< class > class StoragePolicy>  
Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const [inline]
```

Definition at line 109 of file proxy.h.

References Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



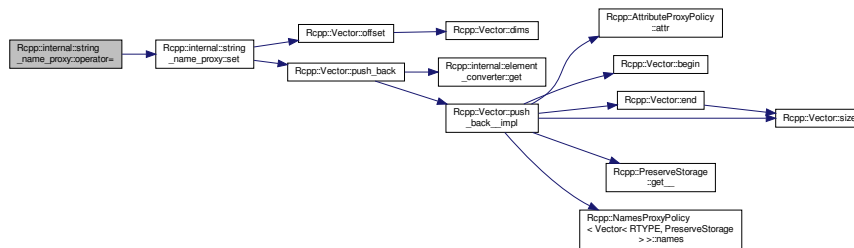
## 6.745.4.6 operator=() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy>
string_name_proxy& Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator= (
    const std::string & rhs ) [inline]
```

Definition at line 96 of file proxy.h.

References Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



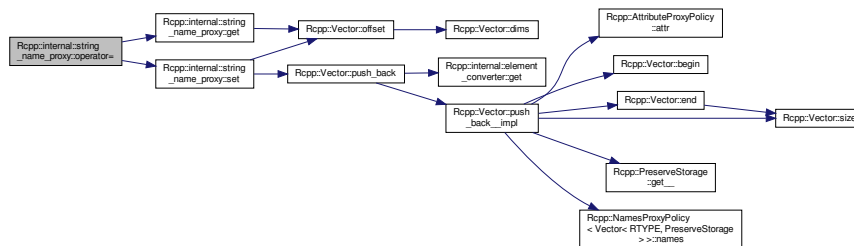
## 6.745.4.7 operator=() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy>
string_name_proxy& Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator= (
    const string_name_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 100 of file proxy.h.

References Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



### 6.745.4.8 operator[]()

```
template<int RTYPE, template< class > class StoragePolicy>
reference Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::operator[] (
    R_xlen_t i ) [inline]
```

Definition at line 115 of file proxy.h.

References Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



### 6.745.4.9 set()

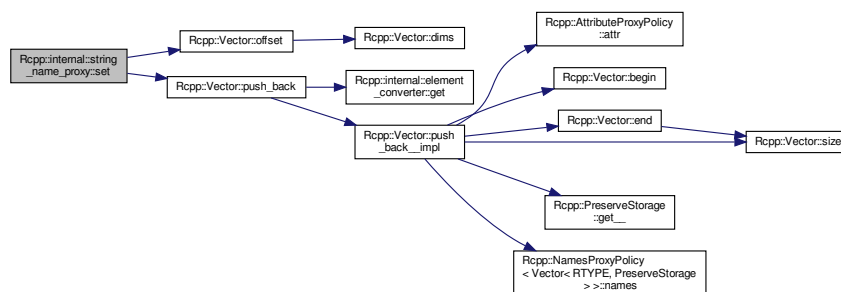
```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::set (
    const std::string & rhs ) [inline], [private]
```

Definition at line 121 of file proxy.h.

References Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::name, Rcpp::Vector< RTYPE, StoragePolicy >::offset(), Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::parent, and Rcpp::Vector< RTYPE, StoragePolicy >::push\_back().

Referenced by Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::operator=().

Here is the call graph for this function:



### 6.745.4.10 size()

```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::size ( ) [inline]
```

Definition at line 116 of file proxy.h.

References Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::get().

Referenced by Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



## 6.745.5 Member Data Documentation

### 6.745.5.1 name

```
template<int RTYPE, template< class > class StoragePolicy>
std::string Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::name [private]
```

Definition at line 120 of file proxy.h.

Referenced by Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::set().

### 6.745.5.2 parent

```
template<int RTYPE, template< class > class StoragePolicy>
VECTOR& Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::parent [private]
```

Definition at line 119 of file proxy.h.

Referenced by Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::set().

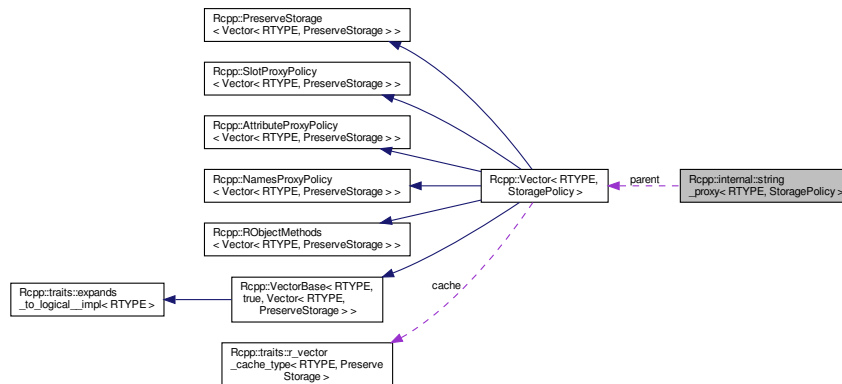
The documentation for this class was generated from the following files:

- inst/include/Rcpp/vector/00\_forward\_proxy.h
- inst/include/Rcpp/vector/proxy.h

## 6.746 Rcpp::internal::string\_proxy< RTYPE, StoragePolicy > Class Template Reference

```
#include <string_proxy.h>
```

Collaboration diagram for Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >:



### Public Types

- typedef `::Rcpp::Vector< RTYPE, StoragePolicy >` `VECTOR`
- typedef const char \* `iterator`
- typedef const char & `reference`

### Public Member Functions

- `string_proxy ()`
- `string_proxy (VECTOR &v, R_xlen_t index_)`
- `string_proxy (const string_proxy &other)`
- `string_proxy & operator= (const string_proxy< RTYPE, StoragePolicy > &other)`
- `template<template< class > class StoragePolicy2>`  
`string_proxy & operator= (const string_proxy< RTYPE, StoragePolicy2 > &other)`
- `template<template< class > class StoragePolicy2>`  
`string_proxy & operator= (const const_string_proxy< RTYPE, StoragePolicy2 > &other)`
- `string_proxy & operator= (const String &s)`
- `template<typename T >`  
`string_proxy & operator= (const std::basic_string< T > &rhs)`
- `string_proxy & operator= (const char *rhs)`
- `string_proxy & operator= (const wchar_t *rhs)`
- `string_proxy & operator= (SEXP rhs)`
- `void import (const string_proxy &other)`
- `template<typename T >`  
`string_proxy & operator+= (const T &rhs)`



- [operator SEXP](#) () const
- [operator char \\*](#) () const
- void [swap](#) ([string\\_proxy](#) &other)
- void [move](#) (R\_xlen\_t n)
- [SEXP get](#) () const
- template<typename T >  
void [set](#) (const T &x)
- void [set](#) (SEXP x)
- [iterator begin](#) () const
- [iterator end](#) () const
- R\_xlen\_t [size](#) () const
- bool [empty](#) () const
- [reference operator\[\]](#) (R\_xlen\_t n)
- template<typename UnaryOperator >  
void [transform](#) (UnaryOperator op)
- template<typename OutputIterator, typename UnaryOperator >  
void [apply](#) (OutputIterator target, UnaryOperator op)
- template<typename UnaryOperator >  
void [apply](#) (UnaryOperator op)
- bool [operator==](#) (const char \*other) const
- bool [operator!=](#) (const char \*other) const
- template<template< class > class SP>  
bool [operator==](#) (const [string\\_proxy](#)< STRSXP, SP > &other) const
- template<template< class > class SP>  
bool [operator!=](#) (const [string\\_proxy](#)< STRSXP, SP > &other) const
- bool [operator==](#) (SEXP other) const
- bool [operator!=](#) (SEXP other) const
- template<typename T >  
[string\\_proxy](#)< RTYPE, StoragePolicy > & [operator+=](#) (const T &rhs)
- template<template< class > class StoragePolicy2>  
[string\\_proxy](#)< RTYPE, StoragePolicy1 > & [operator=](#) (const [const\\_string\\_proxy](#)< RTYPE, StoragePolicy2 > &other)

## Public Attributes

- [VECTOR \\*](#) [parent](#)
- R\_xlen\_t [index](#)

## Static Private Attributes

- static std::string [buffer](#)

## Friends

- template<int RT>  
std::ostream & [operator<<](#) (std::ostream &os, const [string\\_proxy](#)< RT > &proxy)
- template<int RT>  
std::string [operator+](#) (const std::string &x, const [string\\_proxy](#)< RT > &proxy)

### 6.746.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy>  
class Rcpp::internal::string_proxy< RTYPE, StoragePolicy >
```

Definition at line 29 of file string\_proxy.h.

### 6.746.2 Member Typedef Documentation

#### 6.746.2.1 iterator

```
template<int RTYPE, template< class > class StoragePolicy>  
typedef const char* Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::iterator
```

Definition at line 33 of file string\_proxy.h.

#### 6.746.2.2 reference

```
template<int RTYPE, template< class > class StoragePolicy>  
typedef const char& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::reference
```

Definition at line 34 of file string\_proxy.h.

#### 6.746.2.3 VECTOR

```
template<int RTYPE, template< class > class StoragePolicy>  
typedef ::Rcpp::Vector<RTYPE, StoragePolicy> Rcpp::internal::string_proxy< RTYPE, StoragePolicy  
>::VECTOR
```

Definition at line 32 of file string\_proxy.h.

### 6.746.3 Constructor & Destructor Documentation

**6.746.3.1 string\_proxy()** [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::string_proxy ( ) [inline]
```

Definition at line 36 of file string\_proxy.h.

**6.746.3.2 string\_proxy()** [2/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::string_proxy (
    VECTOR & v,
    R_xlen_t index_ ) [inline]
```

Creates a proxy

**Parameters**

<i>v</i>	reference to the associated character vector
<i>index</i>	index

Definition at line 44 of file string\_proxy.h.

**6.746.3.3 string\_proxy()** [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::string_proxy (
    const string_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 46 of file string\_proxy.h.

**6.746.4 Member Function Documentation**

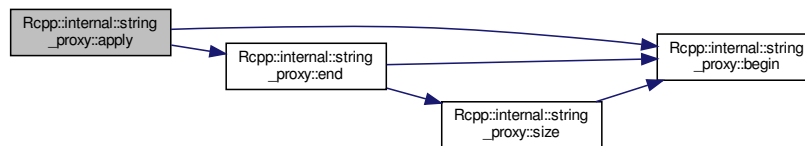
### 6.746.4.1 `apply()` [1/2]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename OutputIterator , typename UnaryOperator >
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::apply (
    OutputIterator target,
    UnaryOperator op ) [inline]
```

Definition at line 181 of file `string_proxy.h`.

References `Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin()`, and `Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::end()`.

Here is the call graph for this function:



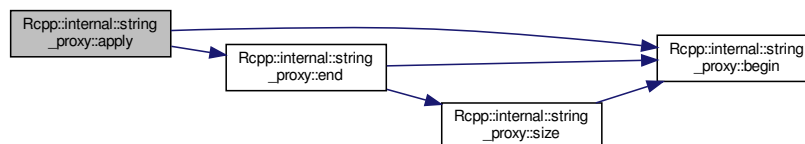
### 6.746.4.2 `apply()` [2/2]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename UnaryOperator >
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::apply (
    UnaryOperator op ) [inline]
```

Definition at line 186 of file `string_proxy.h`.

References `Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin()`, and `Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::end()`.

Here is the call graph for this function:



### 6.746.4.3 begin()

```
template<int RTYPE, template< class > class StoragePolicy>
iterator Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::begin ( ) const [inline]
```

Definition at line 167 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::apply(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::empty(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::end(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::operator!=(), Rcpp::internal::operator<(), Rcpp::internal::operator<=(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::operator==(), Rcpp::internal::operator>(), Rcpp::internal::operator>=(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::operator[](), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::size(), and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::transform().

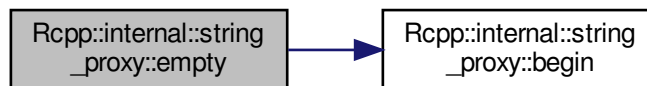
### 6.746.4.4 empty()

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::empty ( ) const [inline]
```

Definition at line 170 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



### 6.746.4.5 end()

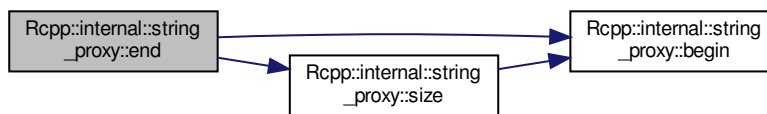
```
template<int RTYPE, template< class > class StoragePolicy>
iterator Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::end ( ) const [inline]
```

Definition at line 168 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin(), and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::apply().

Here is the call graph for this function:



### 6.746.4.6 get()

```
template<int RTYPE, template< class > class StoragePolicy>
SEXPR Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::get ( ) const [inline]
```

Definition at line 156 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::parent.

Referenced by Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::operator char \*(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::operator SEXP(), Rcpp::String::operator!(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::operator!(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::operator=(), Rcpp::String::operator=(), Rcpp::String::operator==( ), and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::operator==( ).

### 6.746.4.7 import()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::import (
    const string_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 102 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::parent.

**6.746.4.8 move()**

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::move (
    R_xlen_t n) [inline]
```

Definition at line 154 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::index.

**6.746.4.9 operator char \*()**

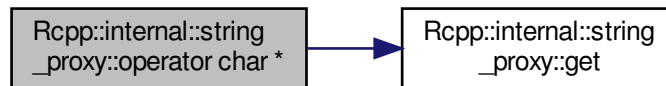
```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator char * ( ) const [inline]
```

rhs use. Retrieves the current value of the element this proxy refers to and convert it to a C string

Definition at line 127 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:

**6.746.4.10 operator SEXP()**

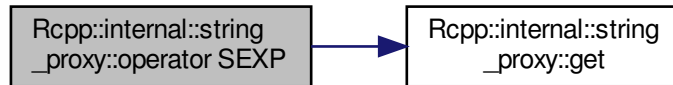
```
template<int RTYPE, template< class > class StoragePolicy>
Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator SEXP ( ) const [inline]
```

rhs use. Retrieves the current value of the element this proxy refers to.

Definition at line 118 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



#### 6.746.4.11 operator"!=() [1/3]

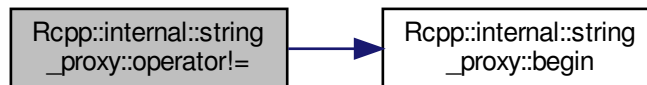
```

template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator!= (
    const char * other ) const [inline]
  
```

Definition at line 193 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



#### 6.746.4.12 operator"!=() [2/3]

```

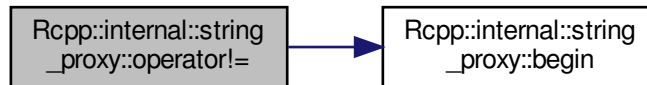
template<int RTYPE, template< class > class StoragePolicy>
template<template< class > class SP>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator!= (
    const string_proxy< STRSXP, SP > & other ) const [inline]
  
```

Definition at line 203 of file string\_proxy.h.



References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



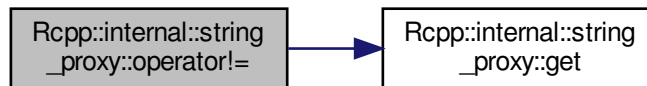
#### 6.746.4.13 operator"!=( ) [3/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator!= (
    SEXP other ) const [inline]
```

Definition at line 211 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



**6.746.4.14 operator+=( ) [1/2]**

```

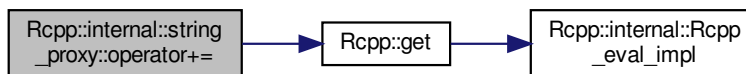
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
string_proxy<RTYPE, StoragePolicy>& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator+=
(
    const T & rhs )

```

Definition at line 700 of file String.h.

References Rcpp::get().

Here is the call graph for this function:

**6.746.4.15 operator+=( ) [2/2]**

```

template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator+= (
    const T & rhs )

```

lhs use. Adds the content of the rhs proxy to the element this proxy refers to.

**6.746.4.16 operator=( ) [1/9]**

```

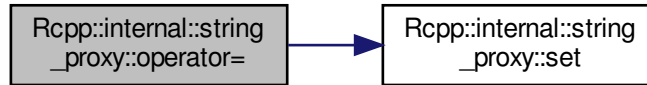
template<int RTYPE, template< class > class StoragePolicy>
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
    const char * rhs ) [inline]

```

Definition at line 85 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



#### 6.746.4.17 operator=() [2/9]

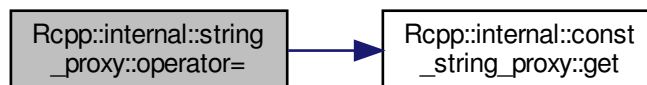
```

template<int RTYPE, template< class > class StoragePolicy>
template<template< class > class StoragePolicy2>
string_proxy<RTYPE, StoragePolicy1>& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >←
::operator= (
    const const_string_proxy< RTYPE, StoragePolicy2 > & other )
  
```

Definition at line 174 of file const\_string\_proxy.h.

References Rcpp::internal::const\_string\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:



#### 6.746.4.18 operator=() [3/9]

```

template<int RTYPE, template< class > class StoragePolicy>
template<template< class > class StoragePolicy2>
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
    const const_string_proxy< RTYPE, StoragePolicy2 > & other )
  
```

**6.746.4.19 operator=()** [4/9]

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
    const std::basic_string< T > & rhs ) [inline]
```

lhs use. Assigns the value of the referred element of the character vector

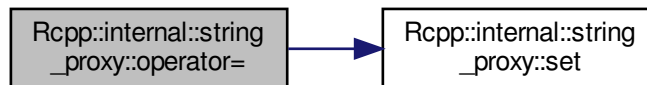
**Parameters**

<i>rhs</i>	new content for the element referred by this proxy
------------	--

Definition at line 80 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:

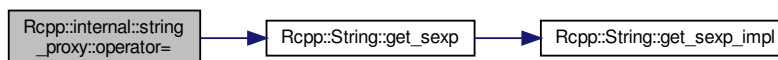
**6.746.4.20 operator=()** [5/9]

```
template<int RTYPE, template< class > class StoragePolicy>
string_proxy< RTYPE, StoragePolicy > & Rcpp::internal::string_proxy< RTYPE, StoragePolicy >↔
::operator= (
    const String & s )
```

Definition at line 682 of file String.h.

References Rcpp::String::get\_sexp().

Here is the call graph for this function:



**6.746.4.21 operator=()** [6/9]

```
template<int RTYPE, template< class > class StoragePolicy>
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
    const string_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

lhs use. Assign the value of the referred element to the current content of the element referred by the rhs proxy

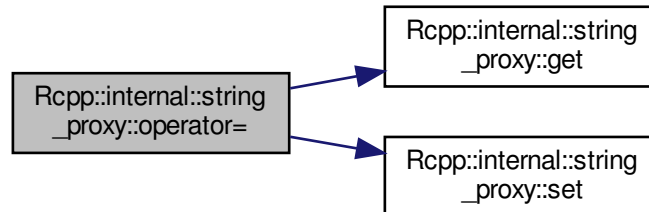
**Parameters**

<i>rhs</i>	another proxy, possibly from another vector
------------	---

Definition at line 57 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:

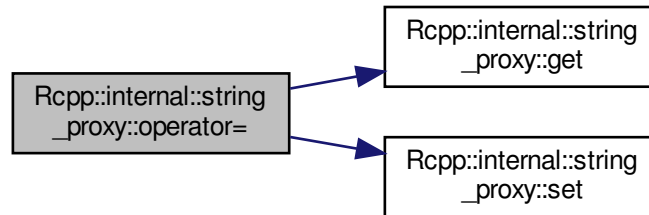
**6.746.4.22 operator=()** [7/9]

```
template<int RTYPE, template< class > class StoragePolicy>
template<template< class > class StoragePolicy2>
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
    const string_proxy< RTYPE, StoragePolicy2 > & other ) [inline]
```

Definition at line 63 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::get(), and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



#### 6.746.4.23 `operator=()` [8/9]

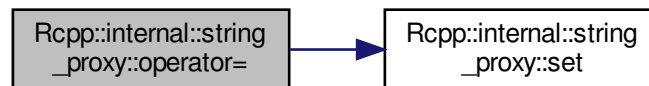
```

template<int RTYPE, template< class > class StoragePolicy>
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
    const wchar_t * rhs ) [inline]
  
```

Definition at line 90 of file `string_proxy.h`.

References `Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set()`.

Here is the call graph for this function:



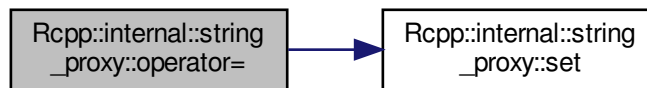
**6.746.4.24 operator=()** [9/9]

```
template<int RTYPE, template< class > class StoragePolicy>
string_proxy& Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator= (
    SEXP rhs ) [inline]
```

Definition at line 96 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:

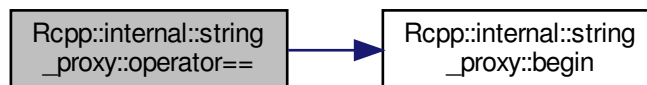
**6.746.4.25 operator==(** [1/3]

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator==(
    const char * other ) const [inline]
```

Definition at line 190 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:



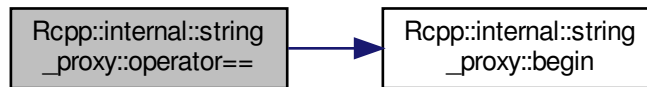
**6.746.4.26 operator==( ) [2/3]**

```
template<int RTYPE, template< class > class StoragePolicy>
template<template< class > class SP>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator==(
    const string_proxy< STRSXP, SP > & other ) const [inline]
```

Definition at line 198 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:

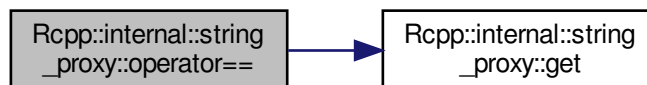
**6.746.4.27 operator==( ) [3/3]**

```
template<int RTYPE, template< class > class StoragePolicy>
bool Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator==(
    SEXP other ) const [inline]
```

Definition at line 207 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::get().

Here is the call graph for this function:





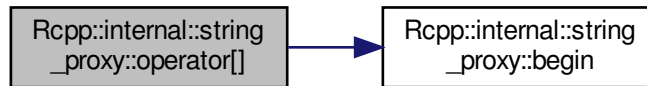
**6.746.4.28 operator[]()**

```
template<int RTYPE, template< class > class StoragePolicy>
reference Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::operator[] (
    R_xlen_t n ) [inline]
```

Definition at line 171 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin().

Here is the call graph for this function:

**6.746.4.29 set() [1/2]**

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename T >
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set (
    const T & x ) [inline]
```

Definition at line 160 of file string\_proxy.h.

Referenced by Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::operator=(), and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::transform().

**6.746.4.30 set() [2/2]**

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::set (
    SEXP x ) [inline]
```

Definition at line 163 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::parent.

### 6.746.4.31 size()

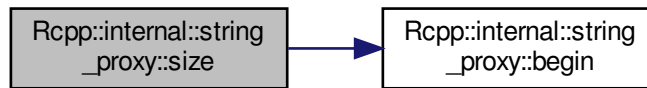
```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::size ( ) const [inline]
```

Definition at line 169 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin().

Referenced by Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::end().

Here is the call graph for this function:



### 6.746.4.32 swap()

```
template<int RTYPE, template< class > class StoragePolicy>
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::swap (
    string_proxy< RTYPE, StoragePolicy > & other ) [inline]
```

Definition at line 146 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::index, and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::parent.

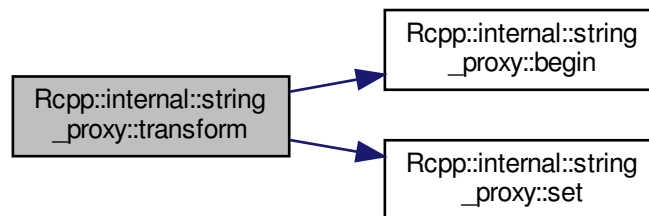
### 6.746.4.33 transform()

```
template<int RTYPE, template< class > class StoragePolicy>
template<typename UnaryOperator >
void Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::transform (
    UnaryOperator op ) [inline]
```

Definition at line 174 of file string\_proxy.h.

References Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::buffer, and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:



## 6.746.5 Friends And Related Function Documentation

### 6.746.5.1 operator+

```
template<int RTYPE, template< class > class StoragePolicy>
template<int RT>
std::string operator+ (
    const std::string & x,
    const string_proxy< RT > & proxy ) [friend]
```

### 6.746.5.2 operator<<

```
template<int RTYPE, template< class > class StoragePolicy>
template<int RT>
std::ostream& operator<< (
    std::ostream & os,
    const string_proxy< RT > & proxy ) [friend]
```

Prints the element this proxy refers to to an output stream

## 6.746.6 Member Data Documentation

### 6.746.6.1 buffer

```
template<int RTYPE, template< class > class StoragePolicy>
std::string Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::buffer [static], [private]
```

Definition at line 216 of file string\_proxy.h.

Referenced by Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::transform().

### 6.746.6.2 index

```
template<int RTYPE, template< class > class StoragePolicy>
R_xlen_t Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::index
```

Definition at line 153 of file string\_proxy.h.

Referenced by Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::get(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::import(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::move(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::set(), and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::swap().

### 6.746.6.3 parent

```
template<int RTYPE, template< class > class StoragePolicy>
VECTOR* Rcpp::internal::string_proxy< RTYPE, StoragePolicy >::parent
```

Definition at line 152 of file string\_proxy.h.

Referenced by Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::begin(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::get(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::import(), Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::set(), and Rcpp::internal::string\_proxy< RTYPE, StoragePolicy >::swap().

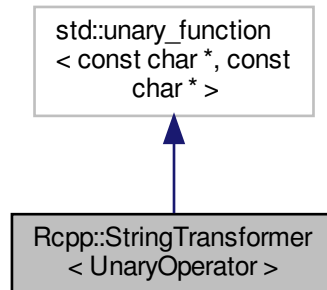
The documentation for this class was generated from the following files:

- [inst/include/Rcpp/vector/00\\_forward\\_proxy.h](#)
- [inst/include/Rcpp/vector/string\\_proxy.h](#)
- [inst/include/Rcpp/String.h](#)
- [inst/include/Rcpp/vector/const\\_string\\_proxy.h](#)

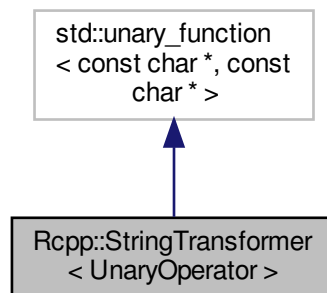
## 6.747 Rcpp::StringTransformer< UnaryOperator > Class Template Reference

```
#include <StringTransformer.h>
```

Inheritance diagram for Rcpp::StringTransformer< UnaryOperator >:



Collaboration diagram for Rcpp::StringTransformer< UnaryOperator >:



### Public Member Functions

- [StringTransformer](#) (const UnaryOperator &op\_)
- [~StringTransformer](#) ()
- const char \* [operator\(\)](#) (const char \*input)

## Private Attributes

- const UnaryOperator & [op](#)
- std::string [buffer](#)

### 6.747.1 Detailed Description

```
template<typename UnaryOperator>
class Rcpp::StringTransformer< UnaryOperator >
```

Definition at line 30 of file StringTransformer.h.

### 6.747.2 Constructor & Destructor Documentation

#### 6.747.2.1 StringTransformer()

```
template<typename UnaryOperator >
Rcpp::StringTransformer< UnaryOperator >::StringTransformer (
    const UnaryOperator & op_ ) [inline]
```

Definition at line 35 of file StringTransformer.h.

#### 6.747.2.2 ~StringTransformer()

```
template<typename UnaryOperator >
Rcpp::StringTransformer< UnaryOperator >::~~StringTransformer ( ) [inline]
```

Definition at line 36 of file StringTransformer.h.

### 6.747.3 Member Function Documentation

#### 6.747.3.1 operator>()()

```
template<typename UnaryOperator >
const char* Rcpp::StringTransformer< UnaryOperator >::operator() (
    const char * input ) [inline]
```

Definition at line 38 of file StringTransformer.h.

References [Rcpp::StringTransformer< UnaryOperator >::buffer](#), and [Rcpp::StringTransformer< UnaryOperator >::op](#).

## 6.747.4 Member Data Documentation

### 6.747.4.1 buffer

```
template<typename UnaryOperator >
std::string Rcpp::StringTransformer< UnaryOperator >::buffer [private]
```

Definition at line 46 of file StringTransformer.h.

Referenced by Rcpp::StringTransformer< UnaryOperator >::operator()().

### 6.747.4.2 op

```
template<typename UnaryOperator >
const UnaryOperator& Rcpp::StringTransformer< UnaryOperator >::op [private]
```

Definition at line 45 of file StringTransformer.h.

Referenced by Rcpp::StringTransformer< UnaryOperator >::operator()().

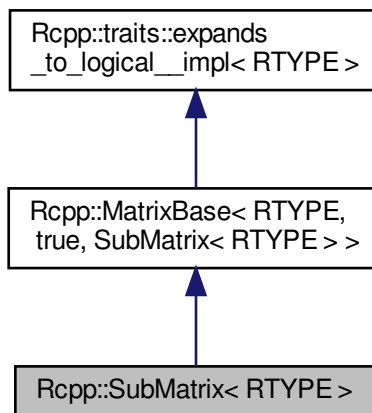
The documentation for this class was generated from the following file:

- inst/include/Rcpp/StringTransformer.h

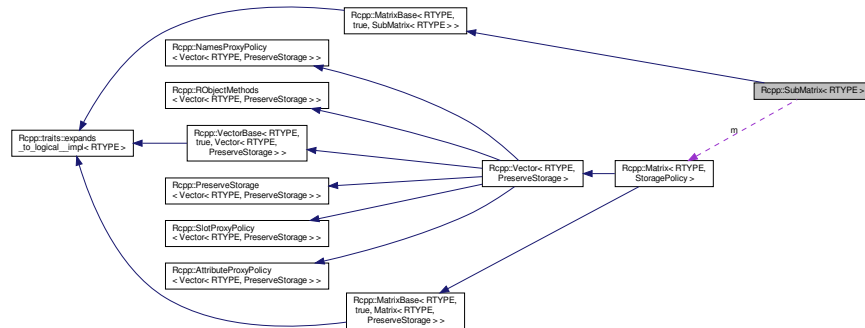
## 6.748 Rcpp::SubMatrix< RTYPE > Class Template Reference

```
#include <SubMatrix.h>
```

Inheritance diagram for Rcpp::SubMatrix< RTYPE >:



Collaboration diagram for `Rcpp::SubMatrix< RTYPE >`:



## Public Types

- typedef `Matrix< RTYPE >` `MATRIX`
- typedef `Vector< RTYPE >::iterator` `vec_iterator`
- typedef `MATRIX::Proxy` `Proxy`

## Public Member Functions

- `SubMatrix (MATRIX &m_, const Range &row_range_, const Range &col_range_)`
- `R_xlen_t size () const`
- `int ncol () const`
- `int nrow () const`
- `Proxy operator() (int i, int j) const`
- `vec_iterator column_iterator (int j) const`

## Private Attributes

- `MATRIX & m`
- `vec_iterator iter`
- `int m_nr`
- `int nc`
- `int nr`

## Additional Inherited Members

### 6.748.1 Detailed Description

```
template<int RTYPE>
class Rcpp::SubMatrix< RTYPE >
```

Definition at line 28 of file `SubMatrix.h`.



## 6.748.2 Member Typedef Documentation

### 6.748.2.1 MATRIX

```
template<int RTYPE>
typedef Matrix<RTYPE> Rcpp::SubMatrix< RTYPE >::MATRIX
```

Definition at line 30 of file SubMatrix.h.

### 6.748.2.2 Proxy

```
template<int RTYPE>
typedef MATRIX::Proxy Rcpp::SubMatrix< RTYPE >::Proxy
```

Definition at line 32 of file SubMatrix.h.

### 6.748.2.3 vec\_iterator

```
template<int RTYPE>
typedef Vector<RTYPE>::iterator Rcpp::SubMatrix< RTYPE >::vec_iterator
```

Definition at line 31 of file SubMatrix.h.

## 6.748.3 Constructor & Destructor Documentation

### 6.748.3.1 SubMatrix()

```
template<int RTYPE>
Rcpp::SubMatrix< RTYPE >::SubMatrix (
    MATRIX & m_,
    const Range & row_range_,
    const Range & col_range_ ) [inline]
```

Definition at line 34 of file SubMatrix.h.

## 6.748.4 Member Function Documentation

### 6.748.4.1 column\_iterator()

```
template<int RTYPE>
vec_iterator Rcpp::SubMatrix< RTYPE >::column_iterator (
    int j ) const [inline]
```

Definition at line 50 of file SubMatrix.h.

References Rcpp::SubMatrix< RTYPE >::iter, and Rcpp::SubMatrix< RTYPE >::m\_nr.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), and Rcpp::Matrix< RTYPE, StoragePolicy >::operator=().

### 6.748.4.2 ncol()

```
template<int RTYPE>
int Rcpp::SubMatrix< RTYPE >::ncol ( ) const [inline]
```

Definition at line 43 of file SubMatrix.h.

References Rcpp::SubMatrix< RTYPE >::nc.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), Rcpp::Matrix< RTYPE, StoragePolicy >::operator=(), and Rcpp::SubMatrix< RTYPE >::size().

### 6.748.4.3 nrow()

```
template<int RTYPE>
int Rcpp::SubMatrix< RTYPE >::nrow ( ) const [inline]
```

Definition at line 44 of file SubMatrix.h.

References Rcpp::SubMatrix< RTYPE >::nr.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::operator=(), and Rcpp::SubMatrix< RTYPE >::size().

#### 6.748.4.4 operator()

```
template<int RTYPE>
Proxy Rcpp::SubMatrix< RTYPE >::operator() (
    int i,
    int j ) const [inline]
```

Definition at line 46 of file SubMatrix.h.

References Rcpp::SubMatrix< RTYPE >::iter, and Rcpp::SubMatrix< RTYPE >::m\_nr.

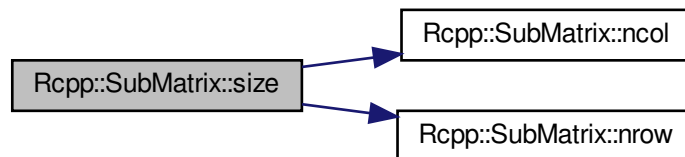
#### 6.748.4.5 size()

```
template<int RTYPE>
R_xlen_t Rcpp::SubMatrix< RTYPE >::size ( ) const [inline]
```

Definition at line 42 of file SubMatrix.h.

References Rcpp::SubMatrix< RTYPE >::ncol(), and Rcpp::SubMatrix< RTYPE >::nrow().

Here is the call graph for this function:



### 6.748.5 Member Data Documentation

#### 6.748.5.1 iter

```
template<int RTYPE>
vec_iterator Rcpp::SubMatrix< RTYPE >::iter [private]
```

Definition at line 54 of file SubMatrix.h.

Referenced by Rcpp::SubMatrix< RTYPE >::column\_iterator(), and Rcpp::SubMatrix< RTYPE >::operator()().

### 6.748.5.2 m

```
template<int RTYPE>
MATRIX& Rcpp::SubMatrix< RTYPE >::m [private]
```

Definition at line 53 of file SubMatrix.h.

### 6.748.5.3 m\_nr

```
template<int RTYPE>
int Rcpp::SubMatrix< RTYPE >::m_nr [private]
```

Definition at line 55 of file SubMatrix.h.

Referenced by `Rcpp::SubMatrix< RTYPE >::column_iterator()`, and `Rcpp::SubMatrix< RTYPE >::operator()()`.

### 6.748.5.4 nc

```
template<int RTYPE>
int Rcpp::SubMatrix< RTYPE >::nc [private]
```

Definition at line 55 of file SubMatrix.h.

Referenced by `Rcpp::SubMatrix< RTYPE >::ncol()`.

### 6.748.5.5 nr

```
template<int RTYPE>
int Rcpp::SubMatrix< RTYPE >::nr [private]
```

Definition at line 55 of file SubMatrix.h.

Referenced by `Rcpp::SubMatrix< RTYPE >::nrow()`.

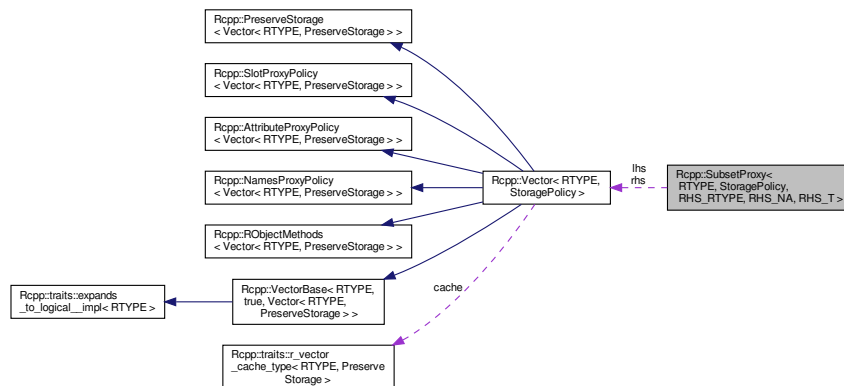
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/vector/SubMatrix.h](#)

## 6.749 Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T > Class Template Reference

```
#include <Subsetter.h>
```

Collaboration diagram for Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >:



### Public Member Functions

- [SubsetProxy](#) (LHS\_t &lhs\_, const RHS\_t &rhs\_)
- [SubsetProxy](#) (const [SubsetProxy](#) &other)
- `template<int OtherRTYPE, template< class > class OtherStoragePolicy>`  
[SubsetProxy](#) & `operator=` (const [Vector](#)< OtherRTYPE, OtherStoragePolicy > &other)
- [SubsetProxy](#) & `operator=` (double other)
- [SubsetProxy](#) & `operator=` (int other)
- [SubsetProxy](#) & `operator=` (const char \*other)
- [SubsetProxy](#) & `operator=` (bool other)
- `template<int RTYPE_OTHER, template< class > class StoragePolicyOther, int RHS_RTYPE_OTHER, bool RHS_NA_OTHER, typename RHS_T_OTHER >`  
[SubsetProxy](#) & `operator=` (const [SubsetProxy](#)< RTYPE\_OTHER, StoragePolicyOther, RHS\_RTYPE\_OTHER, RHS\_NA\_OTHER, RHS\_T\_OTHER > &other)
- [SubsetProxy](#) & `operator=` (const [SubsetProxy](#) &other)
- `operator Vector< RTYPE, StoragePolicy > () const`
- `operator SEXP () const`

### Private Types

- typedef [Vector](#)< RTYPE, StoragePolicy > [LHS\\_t](#)
- typedef [Vector](#)< RHS\_RTYPE, StoragePolicy > [RHS\\_t](#)

## Private Member Functions

- `template<typename IDX >`  
void `check_indices` (IDX \*x, R\_xlen\_t n, R\_xlen\_t size)
- void `get_indices` (traits::identity< traits::int2type< INTSXP > > t)
- void `get_indices` (traits::identity< traits::int2type< REALSXP > > t)
- void `get_indices` (traits::identity< traits::int2type< STRSXP > > t)
- void `get_indices` (traits::identity< traits::int2type< LGLSXP > > t)
- `Vector< RTYPE, StoragePolicy >` `get_vec` () const

## Private Attributes

- `LHS_t` & `lhs`
- const `RHS_t` & `rhs`
- `R_xlen_t` `lhs_n`
- `R_xlen_t` `rhs_n`
- `std::vector< R_xlen_t >` `indices`
- `R_xlen_t` `indices_n`

### 6.749.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T>
class Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >
```

Definition at line 33 of file Subsetter.h.

### 6.749.2 Member Typedef Documentation

#### 6.749.2.1 LHS\_t

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
typedef Vector<RTYPE, StoragePolicy> Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA,
RHS_T >::LHS_t [private]
```

Definition at line 35 of file Subsetter.h.

### 6.749.2.2 RHS\_t

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
typedef Vector<RHS_RTYPE, StoragePolicy> Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE,
RHS_NA, RHS_T >::RHS_t [private]
```

Definition at line 36 of file Subsetter.h.

## 6.749.3 Constructor & Destructor Documentation

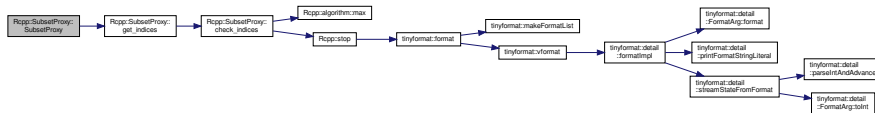
### 6.749.3.1 SubsetProxy() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::SubsetProxy (
    LHS_t & lhs_,
    const RHS_t & rhs_ ) [inline]
```

Definition at line 40 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::get\_indices().

Here is the call graph for this function:



### 6.749.3.2 SubsetProxy() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::SubsetProxy (
    const SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T > & other ) [inline]
```

Definition at line 45 of file Subsetter.h.

## 6.749.4 Member Function Documentation

### 6.749.4.1 check\_indices()

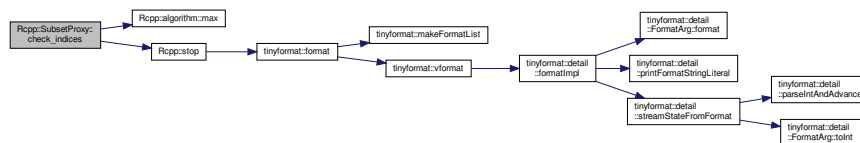
```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTTYPE, bool RHS_NA, typename
RHS_T >
template<typename IDX >
void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTTYPE, RHS_NA, RHS_T >::check_indices (
    IDX * x,
    R_xlen_t n,
    R_xlen_t size ) [inline], [private]
```

Definition at line 138 of file Subsetter.h.

References Rcpp::algorithm::max(), and Rcpp::stop().

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTTYPE, RHS\_NA, RHS\_T >::get\_indices().

Here is the call graph for this function:



### 6.749.4.2 get\_indices() [1/4]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTTYPE, bool RHS_NA, typename
RHS_T >
void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTTYPE, RHS_NA, RHS_T >::get_indices (
    traits::identity< traits::int2type< INTSXP > > t ) [inline], [private]
```

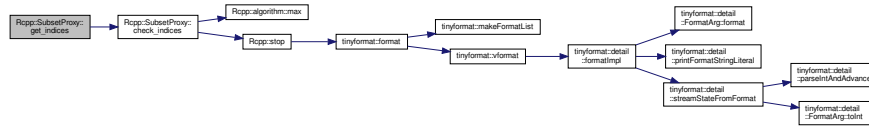
Definition at line 153 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTTYPE, RHS\_NA, RHS\_T >::check\_indices(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTTYPE, RHS\_NA, RHS\_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTTYPE, RHS\_NA, RHS\_T >::indices\_n, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTTYPE, RHS\_NA, RHS\_T >::lhs\_n, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTTYPE, RHS\_NA, RHS\_T >::rhs, and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTTYPE, RHS\_NA, RHS\_T >::rhs\_n.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTTYPE, RHS\_NA, RHS\_T >::SubsetProxy().



Here is the call graph for this function:



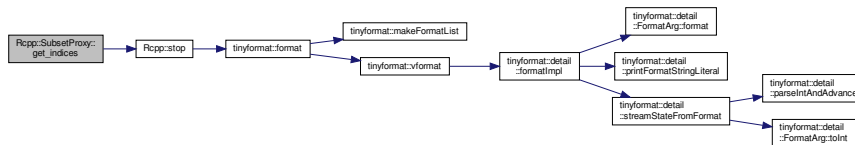
### 6.749.4.3 get\_indices() [2/4]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T >
void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices (
    traits::identity< traits::int2type< LGLSXP > > t ) [inline], [private]
```

Definition at line 191 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::indices\_n, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::lhs\_n, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::rhs, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::rhs\_n, and Rcpp::stop().

Here is the call graph for this function:



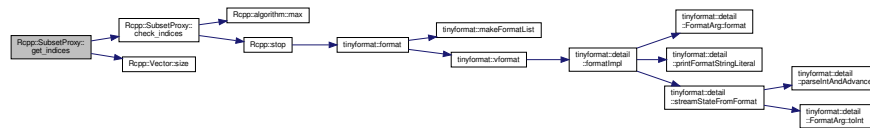
### 6.749.4.4 get\_indices() [3/4]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T >
void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices (
    traits::identity< traits::int2type< REALSXP > > t ) [inline], [private]
```

Definition at line 163 of file Subsetter.h.

References `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::check_indices()`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs_n`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs_n`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



#### 6.749.4.5 `get_indices()` [4/4]

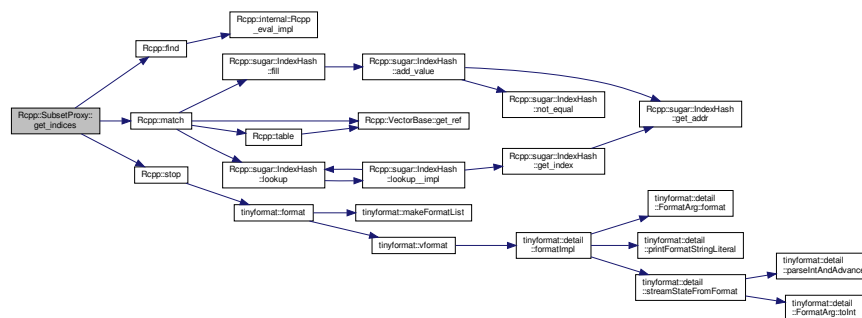
```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename RHS_T >
```

```
void Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices (
    traits::identity< traits::int2type< STRSXP > > t ) [inline], [private]
```

Definition at line 176 of file `Subsetter.h`.

References `Rcpp::find()`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs_n`, `Rcpp::match()`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs_n`, and `Rcpp::stop()`.

Here is the call graph for this function:



## 6.749.4.6 get\_vec()

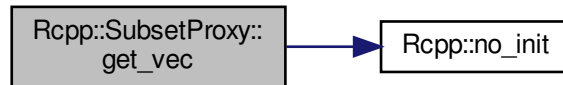
```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
Vector<RTYPE, StoragePolicy> Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_vec ( ) const [inline], [private]
```

Definition at line 208 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::indices\_n, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::lhs, and Rcpp::no\_init().

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::operator SEXP(), and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::operator Vector< RTYPE, StoragePolicy >().

Here is the call graph for this function:



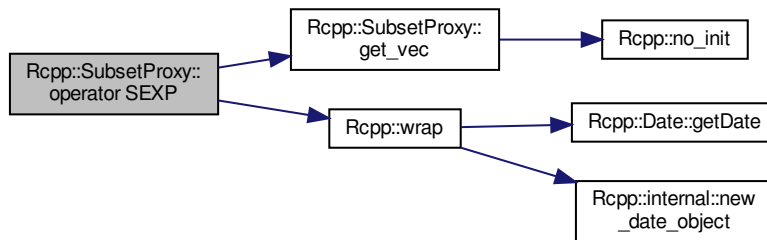
## 6.749.4.7 operator SEXP()

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator SEXP ( ) const
[inline]
```

Definition at line 130 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::get\_vec(), and Rcpp::wrap().

Here is the call graph for this function:



#### 6.749.4.8 operator Vector< RTYPE, StoragePolicy >()

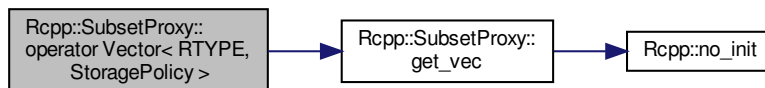
```

template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator Vector< RTYPE,
StoragePolicy > ( ) const [inline]
  
```

Definition at line 126 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::get\_vec().

Here is the call graph for this function:



### 6.749.4.9 operator=( ) [1/7]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
    bool other ) [inline]
```

Definition at line 95 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::indices\_n, and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::lhs.

### 6.749.4.10 operator=( ) [2/7]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
    const char * other ) [inline]
```

Definition at line 88 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::indices\_n, and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::lhs.

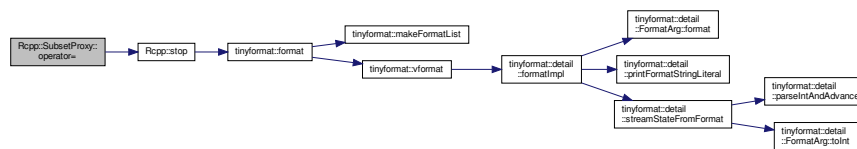
### 6.749.4.11 operator=( ) [3/7]

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
    const SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T > & other ) [inline]
```

Definition at line 110 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::indices\_n, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::lhs, and Rcpp::stop().

Here is the call graph for this function:



**6.749.4.12 operator=()** [4/7]

```

template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
template<int RTYPE_OTHER, template< class > class StoragePolicyOther, int RHS_RTYPE_OTHER, bool
RHS_NA_OTHER, typename RHS_T_OTHER >
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
    const SubsetProxy< RTYPE_OTHER, StoragePolicyOther, RHS_RTYPE_OTHER, RHS_NA_OTHER,
RHS_T_OTHER > & other ) [inline]

```

Definition at line 103 of file Subsetter.h.

**6.749.4.13 operator=()** [5/7]

```

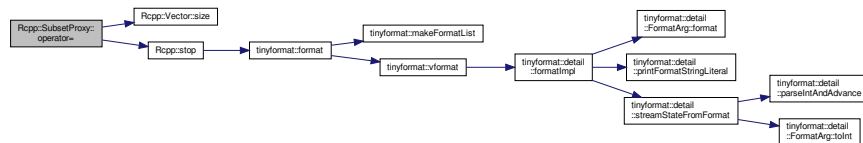
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
template<int OtherRTYPE, template< class > class OtherStoragePolicy>
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
    const Vector< OtherRTYPE, OtherStoragePolicy > & other ) [inline]

```

Definition at line 55 of file Subsetter.h.

References `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs`, `Rcpp::Vector< RTYPE, StoragePolicy >::size()`, and `Rcpp::stop()`.

Here is the call graph for this function:

**6.749.4.14 operator=()** [6/7]

```

template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
    double other ) [inline]

```

Definition at line 74 of file Subsetter.h.

References `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n`, and `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs`.

**6.749.4.15 operator=() [7/7]**

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
SubsetProxy& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator= (
    int other ) [inline]
```

Definition at line 81 of file Subsetter.h.

References Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::indices, Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::indices\_n, and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::lhs.

**6.749.5 Member Data Documentation****6.749.5.1 indices**

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
std::vector<R_xlen_t> Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::
indices [private]
```

Definition at line 230 of file Subsetter.h.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::get\_indices(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::get\_vec(), and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::operator=().

**6.749.5.2 indices\_n**

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
R_xlen_t Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::indices_n [private]
```

Definition at line 233 of file Subsetter.h.

Referenced by Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::get\_indices(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::get\_vec(), and Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::operator=().

### 6.749.5.3 lhs

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
LHS_t& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 225 of file Subsetter.h.

Referenced by `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices()`, `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_vec()`, and `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::operator=()`.

### 6.749.5.4 lhs\_n

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
R_xlen_t Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::lhs_n [private]
```

Definition at line 227 of file Subsetter.h.

Referenced by `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices()`.

### 6.749.5.5 rhs

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
const RHS_t& Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs [private]
```

Definition at line 226 of file Subsetter.h.

Referenced by `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices()`.

### 6.749.5.6 rhs\_n

```
template<int RTYPE, template< class > class StoragePolicy, int RHS_RTYPE, bool RHS_NA, typename
RHS_T >
R_xlen_t Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::rhs_n [private]
```

Definition at line 228 of file Subsetter.h.

Referenced by `Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T >::get_indices()`.

The documentation for this class was generated from the following file:

- `inst/include/Rcpp/vector/Subsetter.h`



## 6.750 `tinyformat::detail::is_convertible< T1, T2 >::succeed` Struct Reference

### Public Attributes

- char `dummy`

### 6.750.1 Detailed Description

```
template<typename T1, typename T2>
struct tinyformat::detail::is_convertible< T1, T2 >::succeed
```

Definition at line 181 of file `tinyformat.h`.

### 6.750.2 Member Data Documentation

#### 6.750.2.1 `dummy`

```
template<typename T1 , typename T2 >
char tinyformat::detail::is_convertible< T1, T2 >::succeed::dummy
```

Definition at line 181 of file `tinyformat.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/utils/tinyformat/tinyformat.h`

## 6.751 `Rcpp::sugar::sugar_const_iterator_type< T >` Struct Template Reference

```
#include <iterator.h>
```

### Public Types

- typedef `SugarIterator< T > type`

### 6.751.1 Detailed Description

```
template<typename T>
struct Rcpp::sugar::sugar_const_iterator_type< T >
```

Definition at line 110 of file iterator.h.

### 6.751.2 Member Typedef Documentation

#### 6.751.2.1 type

```
template<typename T >
typedef SugarIterator<T> Rcpp::sugar::sugar\_const\_iterator\_type< T >::type
```

Definition at line 111 of file iterator.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/tools/iterator.h](#)

## 6.752 Rcpp::sugar::sugar\_const\_iterator\_type< CharacterVector > Struct Reference

```
#include <iterator.h>
```

### Public Types

- typedef SEXP \* [type](#)

### 6.752.1 Detailed Description

Definition at line 116 of file iterator.h.

### 6.752.2 Member Typedef Documentation

### 6.752.2.1 type

```
typedef SEXP* Rcpp::sugar::sugar_const_iterator_type< CharacterVector >::type
```

Definition at line 117 of file iterator.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/tools/iterator.h](#)

## 6.753 Rcpp::sugar::sugar\_const\_iterator\_type< Rcpp::Vector< RTYPE > > Struct Template Reference

```
#include <iterator.h>
```

### Public Types

- typedef [Rcpp::Vector< RTYPE >::const\\_iterator](#) [type](#)

### 6.753.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::sugar::sugar_const_iterator_type< Rcpp::Vector< RTYPE > >
```

Definition at line 113 of file iterator.h.

### 6.753.2 Member Typedef Documentation

#### 6.753.2.1 type

```
template<int RTYPE>
typedef Rcpp::Vector<RTYPE>::const_iterator Rcpp::sugar::sugar_const_iterator_type< Rcpp::Vector<
RTYPE > >::type
```

Definition at line 114 of file iterator.h.

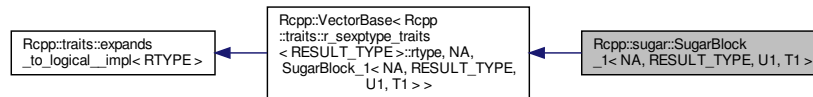
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/tools/iterator.h](#)

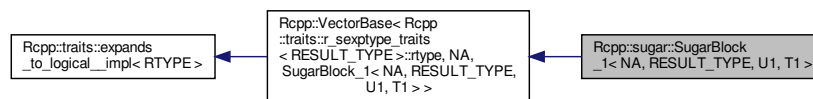
## 6.754 Rcpp::sugar::SugarBlock\_1< NA, RESULT\_TYPE, U1, T1 > Class Template Reference

```
#include <SugarBlock_1.h>
```

Inheritance diagram for Rcpp::sugar::SugarBlock\_1< NA, RESULT\_TYPE, U1, T1 >:



Collaboration diagram for Rcpp::sugar::SugarBlock\_1< NA, RESULT\_TYPE, U1, T1 >:



### Public Types

- typedef RESULT\_TYPE(\* [FunPtr](#)) (U1)

### Public Member Functions

- [SugarBlock\\_1](#) ([FunPtr](#) ptr\_, const T1 &vec\_)
- RESULT\_TYPE [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

### Private Attributes

- [FunPtr](#) ptr
- const T1 & [vec](#)

### 6.754.1 Detailed Description

```
template<bool NA, typename RESULT_TYPE, typename U1, typename T1>
class Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >
```

Definition at line 29 of file SugarBlock\_1.h.

## 6.754.2 Member Typedef Documentation

### 6.754.2.1 FunPtr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 >
typedef RESULT_TYPE(* Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::FunPtr) (U1)
```

Definition at line 31 of file SugarBlock\_1.h.

## 6.754.3 Constructor & Destructor Documentation

### 6.754.3.1 SugarBlock\_1()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 >
Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::SugarBlock_1 (
    FunPtr ptr_,
    const T1 & vec_ ) [inline]
```

Definition at line 32 of file SugarBlock\_1.h.

## 6.754.4 Member Function Documentation

### 6.754.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 >
RESULT_TYPE Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 34 of file SugarBlock\_1.h.

References [Rcpp::sugar::SugarBlock\\_1< NA, RESULT\\_TYPE, U1, T1 >::ptr](#), and [Rcpp::sugar::SugarBlock\\_1< NA, RESULT\\_TYPE, U1, T1 >::vec](#).

### 6.754.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 >
R_xlen_t Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::size ( ) const [inline]
```

Definition at line 37 of file SugarBlock\_1.h.

References Rcpp::sugar::SugarBlock\_1< NA, RESULT\_TYPE, U1, T1 >::vec.

## 6.754.5 Member Data Documentation

### 6.754.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 >
FunPtr Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::ptr [private]
```

Definition at line 40 of file SugarBlock\_1.h.

Referenced by Rcpp::sugar::SugarBlock\_1< NA, RESULT\_TYPE, U1, T1 >::operator[]().

### 6.754.5.2 vec

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 >
const T1& Rcpp::sugar::SugarBlock_1< NA, RESULT_TYPE, U1, T1 >::vec [private]
```

Definition at line 41 of file SugarBlock\_1.h.

Referenced by Rcpp::sugar::SugarBlock\_1< NA, RESULT\_TYPE, U1, T1 >::operator[](), and Rcpp::sugar::SugarBlock\_1< NA, RESULT\_TYPE, U1, T1 >::size().

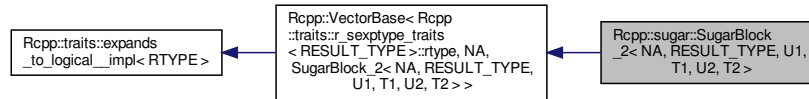
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/SugarBlock\_1.h

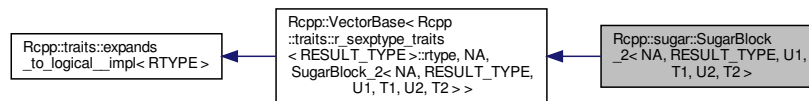
## 6.755 Rcpp::sugar::SugarBlock\_2< NA, RESULT\_TYPE, U1, T1, U2, T2 > Class Template Reference

```
#include <SugarBlock_2.h>
```

Inheritance diagram for Rcpp::sugar::SugarBlock\_2< NA, RESULT\_TYPE, U1, T1, U2, T2 >:



Collaboration diagram for Rcpp::sugar::SugarBlock\_2< NA, RESULT\_TYPE, U1, T1, U2, T2 >:



### Public Types

- typedef RESULT\_TYPE(\* [FunPtr](#)) (U1, U2)

### Public Member Functions

- [SugarBlock\\_2](#) ([FunPtr](#) ptr\_, const T1 &x\_, const T2 &y\_)
- RESULT\_TYPE [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

### Private Attributes

- [FunPtr](#) ptr
- const T1 &x
- const T2 &y

#### 6.755.1 Detailed Description

```
template<bool NA, typename RESULT_TYPE, typename U1, typename T1, typename U2, typename T2>
class Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >
```

Definition at line 29 of file SugarBlock\_2.h.

## 6.755.2 Member Typedef Documentation

### 6.755.2.1 FunPtr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >
typedef RESULT_TYPE(* Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::FunPtr) (U1,
U2)
```

Definition at line 31 of file SugarBlock\_2.h.

## 6.755.3 Constructor & Destructor Documentation

### 6.755.3.1 SugarBlock\_2()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >
Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::SugarBlock_2 (
    FunPtr ptr_,
    const T1 & x_,
    const T2 & y_ ) [inline]
```

Definition at line 32 of file SugarBlock\_2.h.

## 6.755.4 Member Function Documentation

### 6.755.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >
RESULT_TYPE Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 37 of file SugarBlock\_2.h.

References [Rcpp::sugar::SugarBlock\\_2< NA, RESULT\\_TYPE, U1, T1, U2, T2 >::ptr](#), [Rcpp::sugar::SugarBlock\\_2< NA, RESULT\\_TYPE, U1, T1, U2, T2 >::x](#), and [Rcpp::sugar::SugarBlock\\_2< NA, RESULT\\_TYPE, U1, T1, U2, T2 >::y](#).



### 6.755.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >  
R_xlen_t Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::size ( ) const [inline]
```

Definition at line 40 of file SugarBlock\_2.h.

References Rcpp::sugar::SugarBlock\_2< NA, RESULT\_TYPE, U1, T1, U2, T2 >::x.

## 6.755.5 Member Data Documentation

### 6.755.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >  
FunPtr Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::ptr [private]
```

Definition at line 43 of file SugarBlock\_2.h.

Referenced by Rcpp::sugar::SugarBlock\_2< NA, RESULT\_TYPE, U1, T1, U2, T2 >::operator[ ]().

### 6.755.5.2 x

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >  
const T1& Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::x [private]
```

Definition at line 44 of file SugarBlock\_2.h.

Referenced by Rcpp::sugar::SugarBlock\_2< NA, RESULT\_TYPE, U1, T1, U2, T2 >::operator[ ](), and Rcpp::sugar::SugarBlock\_2< NA, RESULT\_TYPE, U1, T1, U2, T2 >::size().

### 6.755.5.3 y

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 >  
const T2& Rcpp::sugar::SugarBlock_2< NA, RESULT_TYPE, U1, T1, U2, T2 >::y [private]
```

Definition at line 45 of file SugarBlock\_2.h.

Referenced by Rcpp::sugar::SugarBlock\_2< NA, RESULT\_TYPE, U1, T1, U2, T2 >::operator[ ]().

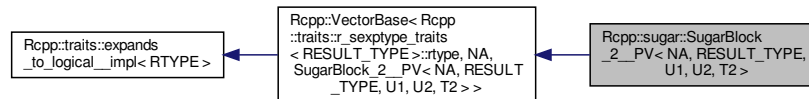
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/block/SugarBlock\\_2.h](#)

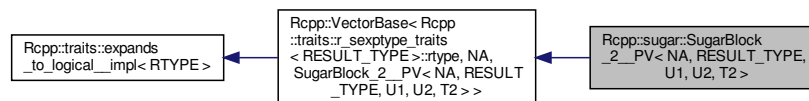
## 6.756 Rcpp::sugar::SugarBlock\_2\_\_PV< NA, RESULT\_TYPE, U1, U2, T2 > Class Template Reference

```
#include <SugarBlock_2.h>
```

Inheritance diagram for Rcpp::sugar::SugarBlock\_2\_\_PV< NA, RESULT\_TYPE, U1, U2, T2 >:



Collaboration diagram for Rcpp::sugar::SugarBlock\_2\_\_PV< NA, RESULT\_TYPE, U1, U2, T2 >:



### Public Types

- typedef RESULT\_TYPE(\* FunPtr) (U1, U2)

### Public Member Functions

- SugarBlock\_2\_\_PV (FunPtr ptr\_, U1 u1, const T2 &y\_)
- RESULT\_TYPE operator[] (R\_xlen\_t i) const
- R\_xlen\_t size () const

### Private Attributes

- FunPtr ptr
- U1 x
- const T2 & y

#### 6.756.1 Detailed Description

```
template<bool NA, typename RESULT_TYPE, typename U1, typename U2, typename T2>
class Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >
```

Definition at line 68 of file SugarBlock\_2.h.

## 6.756.2 Member Typedef Documentation

### 6.756.2.1 FunPtr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >
typedef RESULT_TYPE(* Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::FunPtr) (U1,
U2)
```

Definition at line 70 of file SugarBlock\_2.h.

## 6.756.3 Constructor & Destructor Documentation

### 6.756.3.1 SugarBlock\_2\_\_PV()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >
Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::SugarBlock_2__PV (
    FunPtr ptr_,
    U1 u1,
    const T2 & y_ ) [inline]
```

Definition at line 71 of file SugarBlock\_2.h.

## 6.756.4 Member Function Documentation

### 6.756.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >
RESULT_TYPE Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 74 of file SugarBlock\_2.h.

References [Rcpp::sugar::SugarBlock\\_2\\_\\_PV< NA, RESULT\\_TYPE, U1, U2, T2 >::ptr](#), [Rcpp::sugar::SugarBlock\\_2\\_\\_PV< NA, RESULT\\_TYPE, U1, U2, T2 >::x](#), and [Rcpp::sugar::SugarBlock\\_2\\_\\_PV< NA, RESULT\\_TYPE, U1, U2, T2 >::y](#).

### 6.756.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >  
R_xlen_t Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::size ( ) const [inline]
```

Definition at line 77 of file SugarBlock\_2.h.

References Rcpp::sugar::SugarBlock\_2\_\_PV< NA, RESULT\_TYPE, U1, U2, T2 >::y.

## 6.756.5 Member Data Documentation

### 6.756.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >  
FunPtr Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::ptr [private]
```

Definition at line 80 of file SugarBlock\_2.h.

Referenced by Rcpp::sugar::SugarBlock\_2\_\_PV< NA, RESULT\_TYPE, U1, U2, T2 >::operator[]().

### 6.756.5.2 x

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >  
U1 Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::x [private]
```

Definition at line 81 of file SugarBlock\_2.h.

Referenced by Rcpp::sugar::SugarBlock\_2\_\_PV< NA, RESULT\_TYPE, U1, U2, T2 >::operator[]().

### 6.756.5.3 y

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename U2 , typename T2 >  
const T2& Rcpp::sugar::SugarBlock_2__PV< NA, RESULT_TYPE, U1, U2, T2 >::y [private]
```

Definition at line 82 of file SugarBlock\_2.h.

Referenced by Rcpp::sugar::SugarBlock\_2\_\_PV< NA, RESULT\_TYPE, U1, U2, T2 >::operator[](), and Rcpp::sugar::SugarBlock\_2\_\_PV< NA, RESULT\_TYPE, U1, U2, T2 >::size().

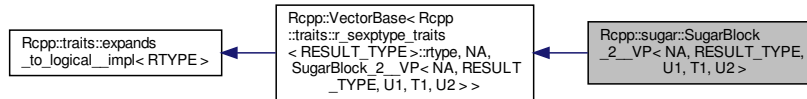
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/block/SugarBlock\\_2.h](#)

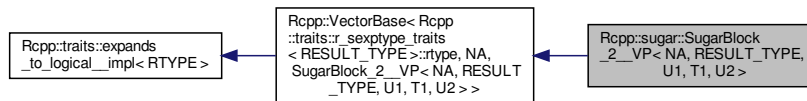
## 6.757 Rcpp::sugar::SugarBlock\_2\_\_VP< NA, RESULT\_TYPE, U1, T1, U2 > Class Template Reference

```
#include <SugarBlock_2.h>
```

Inheritance diagram for Rcpp::sugar::SugarBlock\_2\_\_VP< NA, RESULT\_TYPE, U1, T1, U2 >:



Collaboration diagram for Rcpp::sugar::SugarBlock\_2\_\_VP< NA, RESULT\_TYPE, U1, T1, U2 >:



### Public Types

- typedef RESULT\_TYPE(\* FunPtr) (U1, U2)

### Public Member Functions

- SugarBlock\_2\_\_VP (FunPtr ptr\_, const T1 &x\_, U2 u2)
- RESULT\_TYPE operator[] (R\_xlen\_t i) const
- R\_xlen\_t size () const

### Private Attributes

- FunPtr ptr
- const T1 &x
- U2 y

### 6.757.1 Detailed Description

```
template<bool NA, typename RESULT_TYPE, typename U1, typename T1, typename U2>
class Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >
```

Definition at line 50 of file SugarBlock\_2.h.

## 6.757.2 Member Typedef Documentation

### 6.757.2.1 FunPtr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >
typedef RESULT_TYPE(* Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::FunPtr) (U1,
U2)
```

Definition at line 52 of file SugarBlock\_2.h.

## 6.757.3 Constructor & Destructor Documentation

### 6.757.3.1 SugarBlock\_2\_\_VP()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >
Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::SugarBlock_2__VP (
    FunPtr ptr_,
    const T1 & x_,
    U2 u2 ) [inline]
```

Definition at line 53 of file SugarBlock\_2.h.

## 6.757.4 Member Function Documentation

### 6.757.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >
RESULT_TYPE Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 56 of file SugarBlock\_2.h.

References [Rcpp::sugar::SugarBlock\\_2\\_\\_VP< NA, RESULT\\_TYPE, U1, T1, U2 >::ptr](#), [Rcpp::sugar::SugarBlock\\_2\\_\\_VP< NA, RESULT\\_TYPE, U1, T1, U2 >::x](#), and [Rcpp::sugar::SugarBlock\\_2\\_\\_VP< NA, RESULT\\_TYPE, U1, T1, U2 >::y](#).

### 6.757.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >  
R_xlen_t Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::size ( ) const [inline]
```

Definition at line 59 of file SugarBlock\_2.h.

References Rcpp::sugar::SugarBlock\_2\_\_VP< NA, RESULT\_TYPE, U1, T1, U2 >::x.

## 6.757.5 Member Data Documentation

### 6.757.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >  
FunPtr Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::ptr [private]
```

Definition at line 62 of file SugarBlock\_2.h.

Referenced by Rcpp::sugar::SugarBlock\_2\_\_VP< NA, RESULT\_TYPE, U1, T1, U2 >::operator[]().

### 6.757.5.2 x

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >  
const T1& Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::x [private]
```

Definition at line 63 of file SugarBlock\_2.h.

Referenced by Rcpp::sugar::SugarBlock\_2\_\_VP< NA, RESULT\_TYPE, U1, T1, U2 >::operator[](), and Rcpp::sugar::SugarBlock\_2\_\_VP< NA, RESULT\_TYPE, U1, T1, U2 >::size().

### 6.757.5.3 y

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 >  
U2 Rcpp::sugar::SugarBlock_2__VP< NA, RESULT_TYPE, U1, T1, U2 >::y [private]
```

Definition at line 64 of file SugarBlock\_2.h.

Referenced by Rcpp::sugar::SugarBlock\_2\_\_VP< NA, RESULT\_TYPE, U1, T1, U2 >::operator[]().

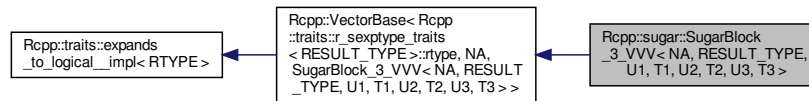
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/block/SugarBlock\\_2.h](#)

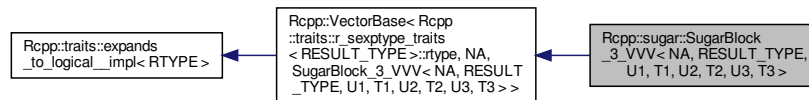
## 6.758 Rcpp::sugar::SugarBlock\_3\_VVV< NA, RESULT\_TYPE, U1, T1, U2, T2, U3, T3 > Class Template Reference

```
#include <SugarBlock_3.h>
```

Inheritance diagram for Rcpp::sugar::SugarBlock\_3\_VVV< NA, RESULT\_TYPE, U1, T1, U2, T2, U3, T3 >:



Collaboration diagram for Rcpp::sugar::SugarBlock\_3\_VVV< NA, RESULT\_TYPE, U1, T1, U2, T2, U3, T3 >:



### Public Types

- typedef RESULT\_TYPE(\* [FunPtr](#)) (U1, U2, U3)

### Public Member Functions

- [SugarBlock\\_3\\_VVV](#) ([FunPtr](#) ptr\_, const T1 &x\_, const T2 &y\_, const T3 &z\_)
- RESULT\_TYPE [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

### Private Attributes

- [FunPtr](#) ptr
- const T1 &x
- const T2 &y
- const T2 &z



## 6.758.1 Detailed Description

```
template<bool NA, typename RESULT_TYPE, typename U1, typename T1, typename U2, typename T2, typename U3, typename T3>
class Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >
```

Definition at line 34 of file SugarBlock\_3.h.

## 6.758.2 Member Typedef Documentation

### 6.758.2.1 FunPtr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
typedef RESULT_TYPE(* Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >↔
::FunPtr) (U1, U2, U3)
```

Definition at line 39 of file SugarBlock\_3.h.

## 6.758.3 Constructor & Destructor Documentation

### 6.758.3.1 SugarBlock\_3\_VVV()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::SugarBlock_3_VVV (
    FunPtr ptr_,
    const T1 & x_,
    const T2 & y_,
    const T3 & z_ ) [inline]
```

Definition at line 40 of file SugarBlock\_3.h.

## 6.758.4 Member Function Documentation

### 6.758.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
RESULT_TYPE Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::operator[]
(
    R_xlen_t i ) const [inline]
```

Definition at line 44 of file SugarBlock\_3.h.

References [Rcpp::sugar::SugarBlock\\_3\\_VVV< NA, RESULT\\_TYPE, U1, T1, U2, T2, U3, T3 >::ptr](#), [Rcpp::sugar::SugarBlock\\_3\\_VVV< NA, RESULT\\_TYPE, U1, T1, U2, T2, U3, T3 >::x](#), [Rcpp::sugar::SugarBlock\\_3\\_VVV< NA, RESULT\\_TYPE, U1, T1, U2, T2, U3, T3 >::y](#), and [Rcpp::sugar::SugarBlock\\_3\\_VVV< NA, RESULT\\_TYPE, U1, T1, U2, T2, U3, T3 >::z](#).

### 6.758.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
R_xlen_t Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::size ( ) const
[inline]
```

Definition at line 47 of file SugarBlock\_3.h.

References [Rcpp::sugar::SugarBlock\\_3\\_VVV< NA, RESULT\\_TYPE, U1, T1, U2, T2, U3, T3 >::x](#).

## 6.758.5 Member Data Documentation

### 6.758.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
FunPtr Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::ptr [private]
```

Definition at line 50 of file SugarBlock\_3.h.

Referenced by [Rcpp::sugar::SugarBlock\\_3\\_VVV< NA, RESULT\\_TYPE, U1, T1, U2, T2, U3, T3 >::operator\[\]\(\)](#).

**6.758.5.2 x**

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
const T1& Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::x [private]
```

Definition at line 51 of file SugarBlock\_3.h.

Referenced by Rcpp::sugar::SugarBlock\_3\_VVV< NA, RESULT\_TYPE, U1, T1, U2, T2, U3, T3 >::operator[](), and Rcpp::sugar::SugarBlock\_3\_VVV< NA, RESULT\_TYPE, U1, T1, U2, T2, U3, T3 >::size().

**6.758.5.3 y**

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
const T2& Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::y [private]
```

Definition at line 52 of file SugarBlock\_3.h.

Referenced by Rcpp::sugar::SugarBlock\_3\_VVV< NA, RESULT\_TYPE, U1, T1, U2, T2, U3, T3 >::operator[]().

**6.758.5.4 z**

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename U2 , typename T2 ,
typename U3 , typename T3 >
const T2& Rcpp::sugar::SugarBlock_3_VVV< NA, RESULT_TYPE, U1, T1, U2, T2, U3, T3 >::z [private]
```

Definition at line 53 of file SugarBlock\_3.h.

Referenced by Rcpp::sugar::SugarBlock\_3\_VVV< NA, RESULT\_TYPE, U1, T1, U2, T2, U3, T3 >::operator[]().

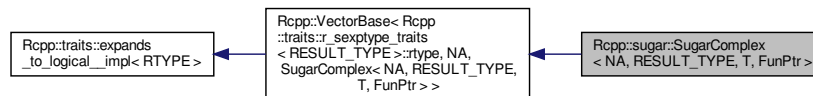
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/block/SugarBlock\\_3.h](#)

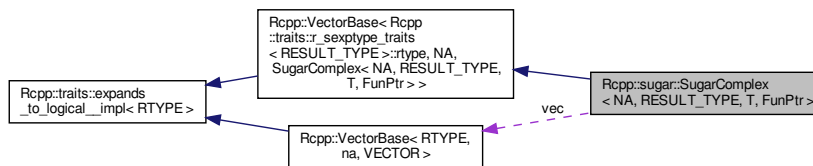
## 6.759 Rcpp::sugar::SugarComplex< NA, RESULT\_TYPE, T, FunPtr > Class Template Reference

```
#include <complex.h>
```

Inheritance diagram for Rcpp::sugar::SugarComplex< NA, RESULT\_TYPE, T, FunPtr >:



Collaboration diagram for Rcpp::sugar::SugarComplex< NA, RESULT\_TYPE, T, FunPtr >:



### Public Types

- typedef [Rcpp::VectorBase< CPLXSXP, NA, T >](#) [VEC\\_TYPE](#)

### Public Member Functions

- [SugarComplex](#) (FunPtr ptr\_, const [VEC\\_TYPE](#) &vec\_)
- [RESULT\\_TYPE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- FunPtr [ptr](#)
- const [VEC\\_TYPE](#) & [vec](#)

### 6.759.1 Detailed Description

```
template<bool NA, typename RESULT_TYPE, typename T, typename FunPtr>
class Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >
```

Definition at line 30 of file complex.h.

## 6.759.2 Member Typedef Documentation

### 6.759.2.1 VEC\_TYPE

```
template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr >
typedef Rcpp::VectorBase<CPLXSCP,NA,T> Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >←
::VEC_TYPE
```

Definition at line 37 of file complex.h.

## 6.759.3 Constructor & Destructor Documentation

### 6.759.3.1 SugarComplex()

```
template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr >
Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::SugarComplex (
    FunPtr ptr_,
    const VEC_TYPE & vec_ ) [inline]
```

Definition at line 39 of file complex.h.

## 6.759.4 Member Function Documentation

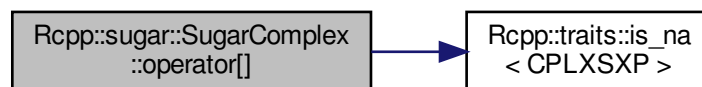
### 6.759.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr >
RESULT_TYPE Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 41 of file complex.h.

References Rcpp::traits::is\_na< CPLXSCP >(), Rcpp::sugar::SugarComplex< NA, RESULT\_TYPE, T, FunPtr >::ptr, and Rcpp::sugar::SugarComplex< NA, RESULT\_TYPE, T, FunPtr >::vec.

Here is the call graph for this function:



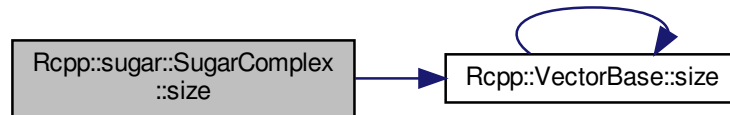
### 6.759.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr >
R_xlen_t Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::size ( ) const [inline]
```

Definition at line 47 of file complex.h.

References `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`, and `Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::vec`.

Here is the call graph for this function:



## 6.759.5 Member Data Documentation

### 6.759.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr >
FunPtr Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::ptr [private]
```

Definition at line 50 of file complex.h.

Referenced by `Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::operator[]()`.

### 6.759.5.2 vec

```
template<bool NA, typename RESULT_TYPE , typename T , typename FunPtr >
const VEC_TYPE& Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::vec [private]
```

Definition at line 51 of file complex.h.

Referenced by `Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::operator[]()`, and `Rcpp::sugar::SugarComplex< NA, RESULT_TYPE, T, FunPtr >::size()`.

The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/complex.h`

## 6.760 Rcpp::sugar::SugarIterator< T > Class Template Reference

```
#include <iterator.h>
```

### Public Types

- typedef `R_xlen_t` `difference_type`
- typedef `Rcpp::traits::storage_type< Rcpp::traits::r_sexptype_traits< T >::rtype >::type` `STORAGE_TYPE`
- typedef `STORAGE_TYPE` `reference`
- typedef `STORAGE_TYPE *` `pointer`
- typedef `std::random_access_iterator_tag` `iterator_category`
- typedef `SugarIterator` `iterator`

### Public Member Functions

- `SugarIterator` (`const T &ref_`)
- `SugarIterator` (`const T &ref_`, `R_xlen_t index_`)
- `SugarIterator` (`const SugarIterator &other`)
- `iterator & operator++` ()
- `iterator operator++` (`int`)
- `iterator & operator--` ()
- `iterator operator--` (`int`)
- `iterator operator+` (`difference_type n`) `const`
- `iterator operator-` (`difference_type n`) `const`
- `iterator & operator+=` (`difference_type n`)
- `iterator & operator-=` (`difference_type n`)
- `reference operator[]` (`R_xlen_t i`)
- `reference operator*` ()
- `pointer operator->` ()
- `bool operator==` (`const iterator &y`) `const`
- `bool operator!=` (`const iterator &y`) `const`
- `bool operator<` (`const iterator &other`) `const`
- `bool operator>` (`const iterator &other`) `const`
- `bool operator<=` (`const iterator &other`) `const`
- `bool operator>=` (`const iterator &other`) `const`
- `difference_type operator-` (`const iterator &other`) `const`

### Private Attributes

- `const T & ref`
- `R_xlen_t index`

### 6.760.1 Detailed Description

```
template<typename T>
class Rcpp::sugar::SugarIterator< T >
```

Definition at line 30 of file `iterator.h`.

## 6.760.2 Member Typedef Documentation

### 6.760.2.1 difference\_type

```
template<typename T >
typedef R_xlen_t Rcpp::sugar::SugarIterator< T >::difference_type
```

Definition at line 33 of file iterator.h.

### 6.760.2.2 iterator

```
template<typename T >
typedef SugarIterator Rcpp::sugar::SugarIterator< T >::iterator
```

Definition at line 38 of file iterator.h.

### 6.760.2.3 iterator\_category

```
template<typename T >
typedef std::random_access_iterator_tag Rcpp::sugar::SugarIterator< T >::iterator_category
```

Definition at line 37 of file iterator.h.

### 6.760.2.4 pointer

```
template<typename T >
typedef STORAGE_TYPE* Rcpp::sugar::SugarIterator< T >::pointer
```

Definition at line 36 of file iterator.h.

### 6.760.2.5 reference

```
template<typename T >
typedef STORAGE_TYPE Rcpp::sugar::SugarIterator< T >::reference
```

Definition at line 35 of file iterator.h.



### 6.760.2.6 STORAGE\_TYPE

```
template<typename T >
typedef Rcpp::traits::storage_type< Rcpp::traits::r_sexptype_traits<T>::rtype >::type Rcpp::sugar::SugarIterator<
T >::STORAGE_TYPE
```

Definition at line 34 of file iterator.h.

## 6.760.3 Constructor & Destructor Documentation

### 6.760.3.1 SugarIterator() [1/3]

```
template<typename T >
Rcpp::sugar::SugarIterator< T >::SugarIterator (
    const T & ref_ ) [inline]
```

Definition at line 40 of file iterator.h.

### 6.760.3.2 SugarIterator() [2/3]

```
template<typename T >
Rcpp::sugar::SugarIterator< T >::SugarIterator (
    const T & ref_,
    R_xlen_t index_ ) [inline]
```

Definition at line 41 of file iterator.h.

### 6.760.3.3 SugarIterator() [3/3]

```
template<typename T >
Rcpp::sugar::SugarIterator< T >::SugarIterator (
    const SugarIterator< T > & other ) [inline]
```

Definition at line 42 of file iterator.h.

## 6.760.4 Member Function Documentation

#### 6.760.4.1 operator"!=(())

```
template<typename T >
bool Rcpp::sugar::SugarIterator< T >::operator!= (
    const iterator & y ) const [inline]
```

Definition at line 84 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

#### 6.760.4.2 operator\*()

```
template<typename T >
reference Rcpp::sugar::SugarIterator< T >::operator* ( ) [inline]
```

Definition at line 74 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index, and Rcpp::sugar::SugarIterator< T >::ref.

#### 6.760.4.3 operator+()

```
template<typename T >
iterator Rcpp::sugar::SugarIterator< T >::operator+ (
    difference_type n ) const [inline]
```

Definition at line 56 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index, and Rcpp::sugar::SugarIterator< T >::ref.

#### 6.760.4.4 operator++() [1/2]

```
template<typename T >
iterator& Rcpp::sugar::SugarIterator< T >::operator++ ( ) [inline]
```

Definition at line 44 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

#### 6.760.4.5 operator++() [2/2]

```
template<typename T >
iterator Rcpp::sugar::SugarIterator< T >::operator++ (
    int ) [inline]
```

Definition at line 45 of file iterator.h.

#### 6.760.4.6 operator+=()

```
template<typename T >
iterator& Rcpp::sugar::SugarIterator< T >::operator+= (
    difference_type n ) [inline]
```

Definition at line 62 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

#### 6.760.4.7 operator-() [1/2]

```
template<typename T >
difference_type Rcpp::sugar::SugarIterator< T >::operator- (
    const iterator & other ) const [inline]
```

Definition at line 100 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

#### 6.760.4.8 operator-() [2/2]

```
template<typename T >
iterator Rcpp::sugar::SugarIterator< T >::operator- (
    difference_type n ) const [inline]
```

Definition at line 59 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index, and Rcpp::sugar::SugarIterator< T >::ref.

#### 6.760.4.9 operator--() [1/2]

```
template<typename T >
iterator& Rcpp::sugar::SugarIterator< T >::operator-- ( ) [inline]
```

Definition at line 50 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

#### 6.760.4.10 operator--() [2/2]

```
template<typename T >
iterator Rcpp::sugar::SugarIterator< T >::operator-- (
    int ) [inline]
```

Definition at line 51 of file iterator.h.

#### 6.760.4.11 operator-=()

```
template<typename T >
iterator& Rcpp::sugar::SugarIterator< T >::operator-= (
    difference_type n ) [inline]
```

Definition at line 66 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

#### 6.760.4.12 operator->()

```
template<typename T >
pointer Rcpp::sugar::SugarIterator< T >::operator-> ( ) [inline]
```

Definition at line 77 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index, and Rcpp::sugar::SugarIterator< T >::ref.

#### 6.760.4.13 operator<()

```
template<typename T >
bool Rcpp::sugar::SugarIterator< T >::operator< (
    const iterator & other ) const [inline]
```

Definition at line 87 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

#### 6.760.4.14 operator<=()

```
template<typename T >
bool Rcpp::sugar::SugarIterator< T >::operator<= (
    const iterator & other ) const [inline]
```

Definition at line 93 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

#### 6.760.4.15 operator==(())

```
template<typename T >
bool Rcpp::sugar::SugarIterator< T >::operator==(
    const iterator & y ) const [inline]
```

Definition at line 81 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

#### 6.760.4.16 operator>()

```
template<typename T >
bool Rcpp::sugar::SugarIterator< T >::operator> (
    const iterator & other ) const [inline]
```

Definition at line 90 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

### 6.760.4.17 operator>=()

```
template<typename T >
bool Rcpp::sugar::SugarIterator< T >::operator>= (
    const iterator & other ) const [inline]
```

Definition at line 96 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index.

### 6.760.4.18 operator[]()

```
template<typename T >
reference Rcpp::sugar::SugarIterator< T >::operator[] (
    R_xlen_t i ) [inline]
```

Definition at line 70 of file iterator.h.

References Rcpp::sugar::SugarIterator< T >::index, and Rcpp::sugar::SugarIterator< T >::ref.

## 6.760.5 Member Data Documentation

### 6.760.5.1 index

```
template<typename T >
R_xlen_t Rcpp::sugar::SugarIterator< T >::index [private]
```

Definition at line 107 of file iterator.h.

Referenced by Rcpp::sugar::SugarIterator< T >::operator!(), Rcpp::sugar::SugarIterator< T >::operator\*(), Rcpp::sugar::SugarIterator< T >::operator+(), Rcpp::sugar::SugarIterator< T >::operator++(), Rcpp::sugar::SugarIterator< T >::operator+=(), Rcpp::sugar::SugarIterator< T >::operator-(), Rcpp::sugar::SugarIterator< T >::operator--(), Rcpp::sugar::SugarIterator< T >::operator-=(), Rcpp::sugar::SugarIterator< T >::operator->(), Rcpp::sugar::SugarIterator< T >::operator<(), Rcpp::sugar::SugarIterator< T >::operator<=(), Rcpp::sugar::SugarIterator< T >::operator==(), Rcpp::sugar::SugarIterator< T >::operator>(), Rcpp::sugar::SugarIterator< T >::operator>=(), and Rcpp::sugar::SugarIterator< T >::operator[]().

## 6.760.5.2 ref

```
template<typename T >
const T& Rcpp::sugar::SugarIterator< T >::ref [private]
```

Definition at line 106 of file iterator.h.

Referenced by Rcpp::sugar::SugarIterator< T >::operator\*(), Rcpp::sugar::SugarIterator< T >::operator+(), Rcpp::sugar::SugarIterator< T >::operator-(), Rcpp::sugar::SugarIterator< T >::operator->(), and Rcpp::sugar::SugarIterator< T >::operator[]().

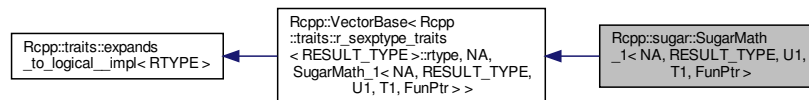
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/tools/iterator.h

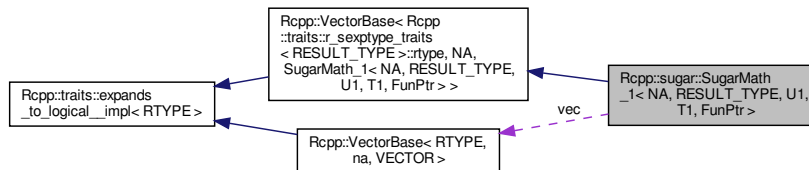
## 6.761 Rcpp::sugar::SugarMath\_1< NA, RESULT\_TYPE, U1, T1, FunPtr > Class Template Reference

```
#include <SugarMath.h>
```

Inheritance diagram for Rcpp::sugar::SugarMath\_1< NA, RESULT\_TYPE, U1, T1, FunPtr >:



Collaboration diagram for Rcpp::sugar::SugarMath\_1< NA, RESULT\_TYPE, U1, T1, FunPtr >:



### Public Types

- typedef Rcpp::VectorBase< Rcpp::traits::r\_sexptype\_traits< RESULT\_TYPE >::rtype, NA, T1 > VEC\_TYPE

## Public Member Functions

- [SugarMath\\_1](#) (FunPtr ptr\_, const [VEC\\_TYPE](#) &vec\_)
- [RESULT\\_TYPE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- FunPtr [ptr](#)
- const [VEC\\_TYPE](#) & [vec](#)

### 6.761.1 Detailed Description

```
template<bool NA, typename RESULT_TYPE, typename U1, typename T1, typename FunPtr>
class Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >
```

Definition at line 29 of file SugarMath.h.

### 6.761.2 Member Typedef Documentation

#### 6.761.2.1 VEC\_TYPE

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename FunPtr >
typedef Rcpp::VectorBase< Rcpp::traits::r_sexptype_traits<RESULT_TYPE>::rtype ,NA,T1> Rcpp::sugar::SugarMath_1<
NA, RESULT_TYPE, U1, T1, FunPtr >::VEC_TYPE
```

Definition at line 36 of file SugarMath.h.

### 6.761.3 Constructor & Destructor Documentation

#### 6.761.3.1 SugarMath\_1()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename FunPtr >
Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::SugarMath_1 (
    FunPtr ptr_,
    const VEC\_TYPE & vec_ ) [inline]
```

Definition at line 38 of file SugarMath.h.



## 6.761.4 Member Function Documentation

### 6.761.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename FunPtr >
RESULT_TYPE Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 40 of file SugarMath.h.

References Rcpp::sugar::SugarMath\_1< NA, RESULT\_TYPE, U1, T1, FunPtr >::ptr, and Rcpp::sugar::SugarMath\_1< NA, RESULT\_TYPE, U1, T1, FunPtr >::vec.

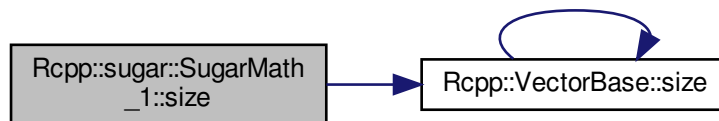
### 6.761.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename FunPtr >
R_xlen_t Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::size ( ) const [inline]
```

Definition at line 45 of file SugarMath.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size(), and Rcpp::sugar::SugarMath\_1< NA, RESULT\_TYPE, U1, T1, FunPtr >::vec.

Here is the call graph for this function:



## 6.761.5 Member Data Documentation

### 6.761.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename FunPtr >
FunPtr Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::ptr [private]
```

Definition at line 48 of file SugarMath.h.

Referenced by `Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::operator[]()`, `Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::operator[]()`, and `Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::operator[]()`.

### 6.761.5.2 vec

```
template<bool NA, typename RESULT_TYPE , typename U1 , typename T1 , typename FunPtr >
const VEC_TYPE& Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec [private]
```

Definition at line 49 of file SugarMath.h.

Referenced by `Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::operator[]()`, `Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::operator[]()`, `Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::operator[]()`, `Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::size()`, `Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::size()`, and `Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::size()`.

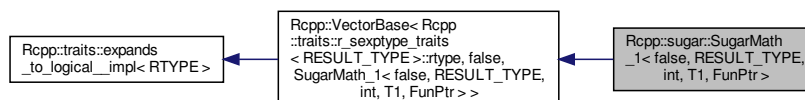
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/block/SugarMath.h](#)

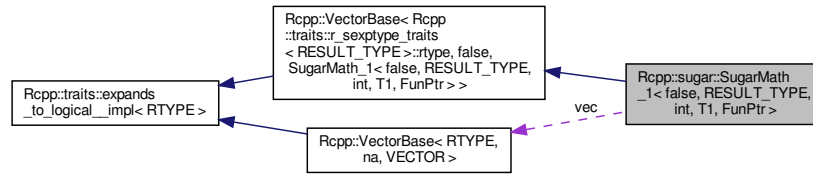
## 6.762 Rcpp::sugar::SugarMath\_1< false, RESULT\_TYPE, int, T1, FunPtr > Class Template Reference

```
#include <SugarMath.h>
```

Inheritance diagram for `Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >`:



Collaboration diagram for Rcpp::sugar::SugarMath\_1< false, RESULT\_TYPE, int, T1, FunPtr >:



## Public Types

- typedef [Rcpp::VectorBase](#)< INTSXP,false, T1 > [VEC\\_TYPE](#)

## Public Member Functions

- [SugarMath\\_1](#) (FunPtr ptr\_, const [VEC\\_TYPE](#) &vec\_)
- [RESULT\\_TYPE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- FunPtr [ptr](#)
- const [VEC\\_TYPE](#) & [vec](#)

### 6.762.1 Detailed Description

```
template<typename RESULT_TYPE, typename T1, typename FunPtr>
class Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >
```

Definition at line 76 of file SugarMath.h.

### 6.762.2 Member Typedef Documentation

#### 6.762.2.1 VEC\_TYPE

```
template<typename RESULT_TYPE , typename T1 , typename FunPtr >
typedef Rcpp::VectorBase< INTSXP ,false,T1> Rcpp::sugar::SugarMath\_1< false, RESULT_TYPE, int,
T1, FunPtr >::VEC_TYPE
```

Definition at line 83 of file SugarMath.h.

## 6.762.3 Constructor & Destructor Documentation

### 6.762.3.1 SugarMath\_1()

```
template<typename RESULT_TYPE , typename T1 , typename FunPtr >
Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::SugarMath_1 (
    FunPtr ptr_,
    const VEC_TYPE & vec_ ) [inline]
```

Definition at line 84 of file SugarMath.h.

## 6.762.4 Member Function Documentation

### 6.762.4.1 operator[]()

```
template<typename RESULT_TYPE , typename T1 , typename FunPtr >
RESULT_TYPE Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 86 of file SugarMath.h.

References `Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::ptr`, and `Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec`.

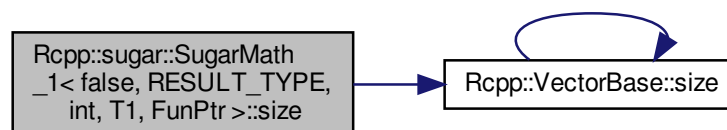
### 6.762.4.2 size()

```
template<typename RESULT_TYPE , typename T1 , typename FunPtr >
R_xlen_t Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::size ( ) const [inline]
```

Definition at line 89 of file SugarMath.h.

References `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`, and `Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec`.

Here is the call graph for this function:



## 6.762.5 Member Data Documentation

### 6.762.5.1 ptr

```
template<typename RESULT_TYPE , typename T1 , typename FunPtr >
FunPtr Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::ptr [private]
```

Definition at line 92 of file SugarMath.h.

### 6.762.5.2 vec

```
template<typename RESULT_TYPE , typename T1 , typename FunPtr >
const VEC_TYPE& Rcpp::sugar::SugarMath_1< false, RESULT_TYPE, int, T1, FunPtr >::vec [private]
```

Definition at line 93 of file SugarMath.h.

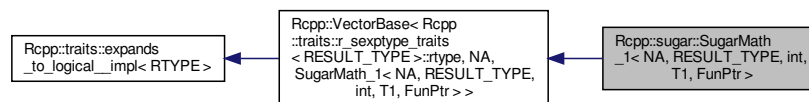
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/block/SugarMath.h](#)

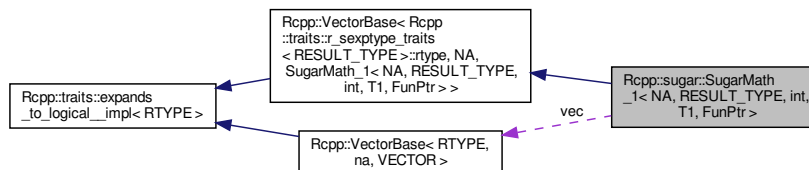
## 6.763 Rcpp::sugar::SugarMath\_1< NA, RESULT\_TYPE, int, T1, FunPtr > Class Template Reference

```
#include <SugarMath.h>
```

Inheritance diagram for Rcpp::sugar::SugarMath\_1< NA, RESULT\_TYPE, int, T1, FunPtr >:



Collaboration diagram for Rcpp::sugar::SugarMath\_1< NA, RESULT\_TYPE, int, T1, FunPtr >:



## Public Types

- typedef [Rcpp::VectorBase](#)< INTSXP,NA, T1 > [VEC\\_TYPE](#)

## Public Member Functions

- [SugarMath\\_1](#) (FunPtr ptr\_, const [VEC\\_TYPE](#) &vec\_)
- [RESULT\\_TYPE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- FunPtr [ptr](#)
- const [VEC\\_TYPE](#) & [vec](#)

### 6.763.1 Detailed Description

```
template<bool NA, typename RESULT_TYPE, typename T1, typename FunPtr>
class Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >
```

Definition at line 53 of file SugarMath.h.

### 6.763.2 Member Typedef Documentation

#### 6.763.2.1 VEC\_TYPE

```
template<bool NA, typename RESULT_TYPE , typename T1 , typename FunPtr >
typedef Rcpp::VectorBase< INTSXP ,NA,T1> Rcpp::sugar::SugarMath\_1< NA, RESULT_TYPE, int, T1,
FunPtr >::VEC_TYPE
```

Definition at line 59 of file SugarMath.h.

### 6.763.3 Constructor & Destructor Documentation

### 6.763.3.1 SugarMath\_1()

```
template<bool NA, typename RESULT_TYPE , typename T1 , typename FunPtr >
Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::SugarMath_1 (
    FunPtr ptr_,
    const VEC_TYPE & vec_ ) [inline]
```

Definition at line 61 of file SugarMath.h.

## 6.763.4 Member Function Documentation

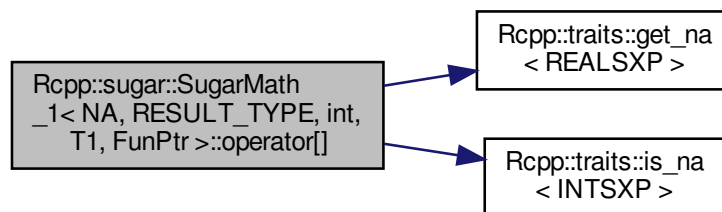
### 6.763.4.1 operator[]()

```
template<bool NA, typename RESULT_TYPE , typename T1 , typename FunPtr >
RESULT_TYPE Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 63 of file SugarMath.h.

References `Rcpp::traits::get_na< REALSXP >()`, `Rcpp::traits::is_na< INTSXP >()`, `Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::ptr`, and `Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec`.

Here is the call graph for this function:



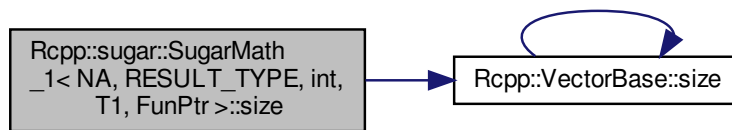
### 6.763.4.2 size()

```
template<bool NA, typename RESULT_TYPE , typename T1 , typename FunPtr >
R_xlen_t Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::size ( ) const [inline]
```

Definition at line 68 of file SugarMath.h.

References `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`, and `Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, U1, T1, FunPtr >::vec`.

Here is the call graph for this function:



## 6.763.5 Member Data Documentation

### 6.763.5.1 ptr

```
template<bool NA, typename RESULT_TYPE , typename T1 , typename FunPtr >
FunPtr Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::ptr [private]
```

Definition at line 71 of file SugarMath.h.

### 6.763.5.2 vec

```
template<bool NA, typename RESULT_TYPE , typename T1 , typename FunPtr >
const VEC_TYPE& Rcpp::sugar::SugarMath_1< NA, RESULT_TYPE, int, T1, FunPtr >::vec [private]
```

Definition at line 72 of file SugarMath.h.

The documentation for this class was generated from the following file:

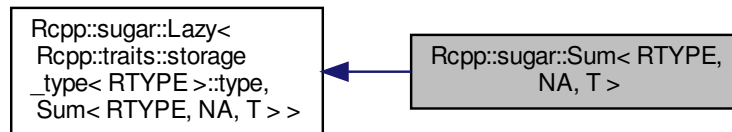
- [inst/include/Rcpp/sugar/block/SugarMath.h](#)



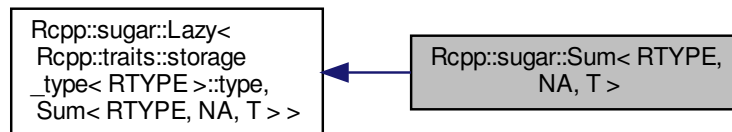
## 6.764 Rcpp::sugar::Sum< RTYPE, NA, T > Class Template Reference

```
#include <sum.h>
```

Inheritance diagram for Rcpp::sugar::Sum< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Sum< RTYPE, NA, T >:



### Public Types

- typedef [Rcpp::VectorBase< RTYPE, NA, T >](#) [VEC\\_TYPE](#)
- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) [STORAGE](#)
- typedef [Rcpp::traits::Extractor< RTYPE, NA, T >::type](#) [VEC\\_EXT](#)

### Public Member Functions

- [Sum](#) (const [VEC\\_TYPE](#) &object\_)
- [STORAGE](#) [get](#) () const

### Private Attributes

- const [VEC\\_EXT](#) & [object](#)

### 6.764.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::Sum< RTYPE, NA, T >
```

Definition at line 29 of file sum.h.

### 6.764.2 Member Typedef Documentation

#### 6.764.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Sum< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file sum.h.

#### 6.764.2.2 VEC\_EXT

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Sum< RTYPE, NA, T >::VEC_EXT
```

Definition at line 33 of file sum.h.

#### 6.764.2.3 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Sum< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file sum.h.

### 6.764.3 Constructor & Destructor Documentation

### 6.764.3.1 Sum()

```
template<int RTYPE, bool NA, typename T >  
Rcpp::sugar::Sum< RTYPE, NA, T >::Sum (  
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 35 of file sum.h.

## 6.764.4 Member Function Documentation

### 6.764.4.1 get()

```
template<int RTYPE, bool NA, typename T >  
STORAGE Rcpp::sugar::Sum< RTYPE, NA, T >::get ( ) const [inline]
```

Definition at line 37 of file sum.h.

## 6.764.5 Member Data Documentation

### 6.764.5.1 object

```
template<int RTYPE, bool NA, typename T >  
const VEC_EXT& Rcpp::sugar::Sum< RTYPE, NA, T >::object [private]
```

Definition at line 50 of file sum.h.

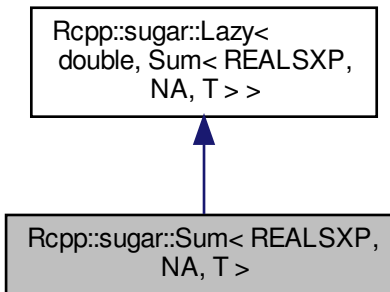
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[sum.h](#)

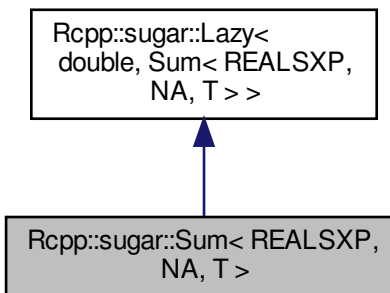
## 6.765 Rcpp::sugar::Sum< REALSXP, NA, T > Class Template Reference

```
#include <sum.h>
```

Inheritance diagram for Rcpp::sugar::Sum< REALSXP, NA, T >:



Collaboration diagram for Rcpp::sugar::Sum< REALSXP, NA, T >:



### Public Types

- typedef [Rcpp::VectorBase< REALSXP, NA, T >](#) [VEC\\_TYPE](#)
- typedef [Rcpp::traits::Extractor< REALSXP, NA, T >::type](#) [VEC\\_EXT](#)

### Public Member Functions

- [Sum](#) (const [VEC\\_TYPE](#) &object\_)
- double [get](#) () const

## Private Attributes

- const [VEC\\_EXT](#) & [object](#)

## 6.765.1 Detailed Description

```
template<bool NA, typename T>
class Rcpp::sugar::Sum< REALSXP, NA, T >
```

Definition at line 54 of file sum.h.

## 6.765.2 Member Typedef Documentation

### 6.765.2.1 VEC\_EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor< REALSXP, NA, T>::type Rcpp::sugar::Sum< REALSXP, NA, T >↔
::VEC_EXT
```

Definition at line 57 of file sum.h.

### 6.765.2.2 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Sum< REALSXP, NA, T >::VEC_TYPE
```

Definition at line 56 of file sum.h.

## 6.765.3 Constructor & Destructor Documentation

### 6.765.3.1 Sum()

```
template<bool NA, typename T >
Rcpp::sugar::Sum< REALSXP, NA, T >::Sum (
    const VEC\_TYPE & object_ ) [inline]
```

Definition at line 59 of file sum.h.

## 6.765.4 Member Function Documentation

### 6.765.4.1 get()

```
template<bool NA, typename T >
double Rcpp::sugar::Sum< REALSXP, NA, T >::get ( ) const [inline]
```

Definition at line 61 of file sum.h.

## 6.765.5 Member Data Documentation

### 6.765.5.1 object

```
template<bool NA, typename T >
const VEC_EXT& Rcpp::sugar::Sum< REALSXP, NA, T >::object [private]
```

Definition at line 70 of file sum.h.

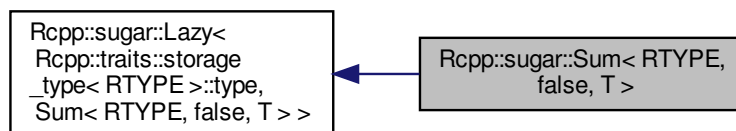
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[sum.h](#)

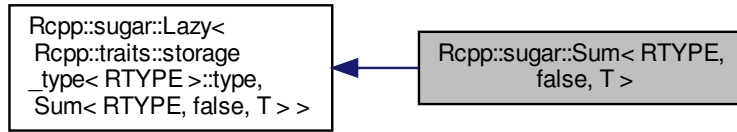
## 6.766 Rcpp::sugar::Sum< RTYPE, false, T > Class Template Reference

```
#include <sum.h>
```

Inheritance diagram for Rcpp::sugar::Sum< RTYPE, false, T >:



Collaboration diagram for Rcpp::sugar::Sum< RTYPE, false, T >:



## Public Types

- typedef [Rcpp::VectorBase< RTYPE, false, T >](#) [VEC\\_TYPE](#)
- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) [STORAGE](#)
- typedef [Rcpp::traits::Extractor< RTYPE, false, T >::type](#) [VEC\\_EXT](#)

## Public Member Functions

- [Sum](#) (const [VEC\\_TYPE](#) &object\_)
- [STORAGE](#) [get](#) () const

## Private Attributes

- const [VEC\\_EXT](#) & [object](#)

### 6.766.1 Detailed Description

```
template<int RTYPE, typename T>
class Rcpp::sugar::Sum< RTYPE, false, T >
```

Definition at line 75 of file sum.h.

### 6.766.2 Member Typedef Documentation

#### 6.766.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::Sum< RTYPE, false, T >::STORAGE
```

Definition at line 78 of file sum.h.

### 6.766.2.2 VEC\_EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor< RTYPE, false, T>::type Rcpp::sugar::Sum< RTYPE, false, T >↔
::VEC_EXT
```

Definition at line 79 of file sum.h.

### 6.766.2.3 VEC\_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Sum< RTYPE, false, T >::VEC_TYPE
```

Definition at line 77 of file sum.h.

## 6.766.3 Constructor & Destructor Documentation

### 6.766.3.1 Sum()

```
template<int RTYPE, typename T >
Rcpp::sugar::Sum< RTYPE, false, T >::Sum (
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 81 of file sum.h.

## 6.766.4 Member Function Documentation

### 6.766.4.1 get()

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Sum< RTYPE, false, T >::get ( ) const [inline]
```

Definition at line 83 of file sum.h.

## 6.766.5 Member Data Documentation



### 6.766.5.1 object

```
template<int RTYPE, typename T >
const VEC_EXT& Rcpp::sugar::Sum< RTYPE, false, T >::object [private]
```

Definition at line 92 of file sum.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/sum.h

## 6.767 Rcpp::SuspendRNGSynchronizationScope Class Reference

```
#include <RNGScope.h>
```

### Public Member Functions

- [SuspendRNGSynchronizationScope \(\)](#)
- [~SuspendRNGSynchronizationScope \(\)](#)

### 6.767.1 Detailed Description

Definition at line 33 of file RNGScope.h.

### 6.767.2 Constructor & Destructor Documentation

#### 6.767.2.1 SuspendRNGSynchronizationScope()

```
Rcpp::SuspendRNGSynchronizationScope::SuspendRNGSynchronizationScope ( ) [inline]
```

Definition at line 35 of file RNGScope.h.

References [Rcpp::internal::beginSuspendRNGSynchronization\(\)](#).

Here is the call graph for this function:



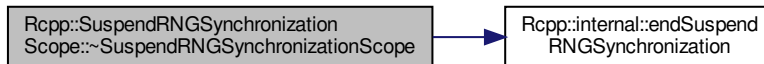
**6.767.2.2 ~SuspendRNGSynchronizationScope()**

```
Rcpp::SuspendRNGSynchronizationScope::~SuspendRNGSynchronizationScope ( ) [inline]
```

Definition at line 36 of file RNGScope.h.

References `Rcpp::internal::endSuspendRNGSynchronization()`.

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- [inst/include/Rcpp/RNGScope.h](#)

**6.768 Rcpp::sugar::Table< RTYPE, TABLE\_T > Class Template Reference**

```
#include <table.h>
```

**Public Types**

- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) [STORAGE](#)

**Public Member Functions**

- [Table](#) (const TABLE\_T &table)
- [operator IntegerVector](#) () const

**Private Types**

- typedef [RCPP\\_UNORDERED\\_MAP< STORAGE, int >](#) [HASH](#)
- typedef [CountInserter< HASH, STORAGE >](#) [Inserter](#)
- typedef [std::map< STORAGE, int, internal::NAComparator< STORAGE > >](#) [SORTED\\_MAP](#)

## Private Attributes

- [HASH](#) hash
- [SORTED\\_MAP](#) map

### 6.768.1 Detailed Description

```
template<int RTYPE, typename TABLE_T>
class Rcpp::sugar::Table< RTYPE, TABLE_T >
```

Definition at line 89 of file table.h.

### 6.768.2 Member Typedef Documentation

#### 6.768.2.1 HASH

```
template<int RTYPE, typename TABLE_T >
typedef RCPP\_UNORDERED\_MAP<STORAGE, int> Rcpp::sugar::Table< RTYPE, TABLE_T >::HASH [private]
```

Definition at line 112 of file table.h.

#### 6.768.2.2 Inserter

```
template<int RTYPE, typename TABLE_T >
typedef CountInserter<HASH,STORAGE> Rcpp::sugar::Table< RTYPE, TABLE_T >::Inserter [private]
```

Definition at line 113 of file table.h.

#### 6.768.2.3 SORTED\_MAP

```
template<int RTYPE, typename TABLE_T >
typedef std::map<STORAGE, int, internal::NAComparator<STORAGE> > Rcpp::sugar::Table< RTYPE,
TABLE_T >::SORTED\_MAP [private]
```

Definition at line 116 of file table.h.

### 6.768.2.4 STORAGE

```
template<int RTYPE, typename TABLE_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Table< RTYPE, TABLE_T >::STORAGE
```

Definition at line 91 of file table.h.

## 6.768.3 Constructor & Destructor Documentation

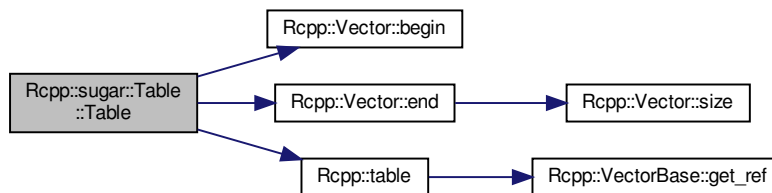
### 6.768.3.1 Table()

```
template<int RTYPE, typename TABLE_T >
Rcpp::sugar::Table< RTYPE, TABLE_T >::Table (
    const TABLE_T & table ) [inline]
```

Definition at line 93 of file table.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::sugar::Table< RTYPE, TABLE_T >::hash`, `Rcpp::sugar::Table< RTYPE, TABLE_T >::map`, and `Rcpp::table()`.

Here is the call graph for this function:



### 6.768.4 Member Function Documentation

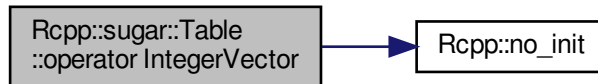
### 6.768.4.1 operator IntegerVector()

```
template<int RTYPE, typename TABLE_T >  
Rcpp::sugar::Table< RTYPE, TABLE_T >::operator IntegerVector ( ) const [inline]
```

Definition at line 101 of file table.h.

References Rcpp::sugar::Table< RTYPE, TABLE\_T >::map, and Rcpp::no\_init().

Here is the call graph for this function:



## 6.768.5 Member Data Documentation

### 6.768.5.1 hash

```
template<int RTYPE, typename TABLE_T >  
HASH Rcpp::sugar::Table< RTYPE, TABLE_T >::hash [private]
```

Definition at line 114 of file table.h.

Referenced by Rcpp::sugar::Table< RTYPE, TABLE\_T >::Table().

### 6.768.5.2 map

```
template<int RTYPE, typename TABLE_T >  
SORTED_MAP Rcpp::sugar::Table< RTYPE, TABLE_T >::map [private]
```

Definition at line 117 of file table.h.

Referenced by Rcpp::sugar::Table< RTYPE, TABLE\_T >::operator IntegerVector(), and Rcpp::sugar::Table< RTYPE, TABLE\_T >::Table().

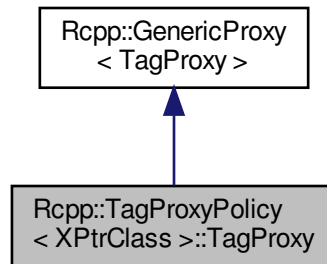
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/[table.h](#)

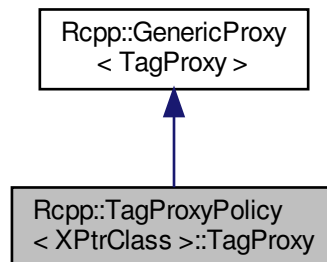
## 6.769 Rcpp::TagProxyPolicy< XPtrClass >::TagProxy Class Reference

```
#include <TagProxy.h>
```

Inheritance diagram for Rcpp::TagProxyPolicy< XPtrClass >::TagProxy:



Collaboration diagram for Rcpp::TagProxyPolicy< XPtrClass >::TagProxy:



### Public Member Functions

- [TagProxy](#) (XPtrClass &xp\_)
- `template<typename U >`  
[TagProxy](#) & `operator=` (const U &u)
- `template<typename U >`  
`operator U` () const
- `operator SEXP` () const
- `template<typename T >`  
[TagProxyPolicy](#)< CLASS >::TagProxy & `operator=` (const T &rhs)

## Private Member Functions

- SEXP [get](#) () const
- void [set](#) (SEXP x)

## Private Attributes

- XPtrClass & [xp](#)

### 6.769.1 Detailed Description

```
template<typename XPtrClass>
class Rcpp::TagProxyPolicy< XPtrClass >::TagProxy
```

Definition at line 27 of file TagProxy.h.

### 6.769.2 Constructor & Destructor Documentation

#### 6.769.2.1 TagProxy()

```
template<typename XPtrClass >
Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::TagProxy (
    XPtrClass & xp_ ) [inline]
```

Definition at line 29 of file TagProxy.h.

### 6.769.3 Member Function Documentation

#### 6.769.3.1 get()

```
template<typename XPtrClass >
SEXP Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::get ( ) const [inline], [private]
```

Definition at line 43 of file TagProxy.h.

References [Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::xp](#).

### 6.769.3.2 operator SEXP()

```
template<typename CLASS >
Rcpp::TagProxyPolicy< CLASS >::TagProxy::operator SEXP
```

Definition at line 111 of file proxy.h.

References Rcpp::get().

Here is the call graph for this function:



### 6.769.3.3 operator U()

```
template<typename XPtrClass >
template<typename U >
Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::operator U ( ) const
```

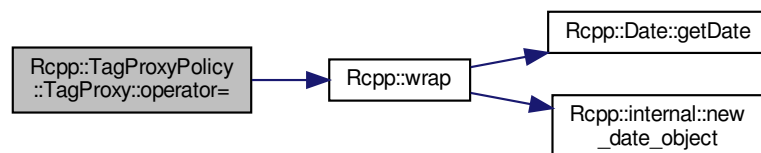
### 6.769.3.4 operator=() [1/2]

```
template<typename XPtrClass >
template<typename T >
TagProxyPolicy<CLASS>::TagProxy& Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::operator= (
    const T & rhs )
```

Definition at line 99 of file proxy.h.

References Rcpp::wrap().

Here is the call graph for this function:





### 6.769.3.5 operator=() [2/2]

```
template<typename XPtrClass >
template<typename U >
TagProxy& Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::operator= (
    const U & u )
```

### 6.769.3.6 set()

```
template<typename XPtrClass >
void Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::set (
    SEXP x ) [inline], [private]
```

Definition at line 47 of file TagProxy.h.

References Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::xp.

## 6.769.4 Member Data Documentation

### 6.769.4.1 xp

```
template<typename XPtrClass >
XPtrClass& Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::xp [private]
```

Definition at line 41 of file TagProxy.h.

Referenced by Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::get(), and Rcpp::TagProxyPolicy< XPtrClass >::TagProxy::set().

The documentation for this class was generated from the following files:

- [inst/include/Rcpp/proxy/TagProxy.h](#)
- [inst/include/Rcpp/api/meat/proxy.h](#)

## 6.770 Rcpp::TagProxyPolicy< XPtrClass > Class Template Reference

```
#include <TagProxy.h>
```

## Classes

- class [const\\_TagProxy](#)
- class [TagProxy](#)

## Public Member Functions

- [TagProxy](#) tag ()
- [const\\_TagProxy](#) tag () const

### 6.770.1 Detailed Description

```
template<typename XPtrClass>  
class Rcpp::TagProxyPolicy< XPtrClass >
```

Definition at line 24 of file TagProxy.h.

### 6.770.2 Member Function Documentation

#### 6.770.2.1 tag() [1/2]

```
template<typename XPtrClass >  
TagProxy Rcpp::TagProxyPolicy< XPtrClass >::tag ( ) [inline]
```

Definition at line 73 of file TagProxy.h.

#### 6.770.2.2 tag() [2/2]

```
template<typename XPtrClass >  
const\_TagProxy Rcpp::TagProxyPolicy< XPtrClass >::tag ( ) const [inline]
```

Definition at line 77 of file TagProxy.h.

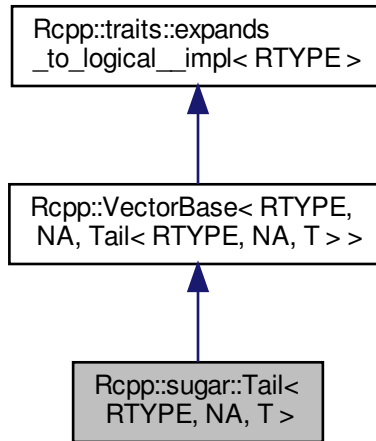
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/proxy/TagProxy.h](#)

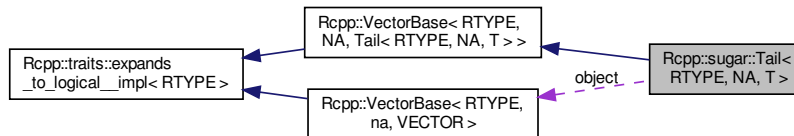
## 6.771 Rcpp::sugar::Tail< RTYPE, NA, T > Class Template Reference

```
#include <tail.h>
```

Inheritance diagram for Rcpp::sugar::Tail< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Tail< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`

### Public Member Functions

- `Tail` (const `VEC_TYPE` &object\_, `R_xlen_t` n\_)
- `STORAGE` `operator[]` (`R_xlen_t` i) const
- `R_xlen_t` `size` () const

## Private Attributes

- const [VEC\\_TYPE](#) & *object*
- [R\\_xlen\\_t](#) *start*
- [R\\_xlen\\_t](#) *n*

### 6.771.1 Detailed Description

```
template<int RTYPE, bool NA, typename T >
class Rcpp::sugar::Tail< RTYPE, NA, T >
```

Definition at line 29 of file tail.h.

### 6.771.2 Member Typedef Documentation

#### 6.771.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Tail< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file tail.h.

#### 6.771.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Tail< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file tail.h.

### 6.771.3 Constructor & Destructor Documentation

#### 6.771.3.1 Tail()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Tail< RTYPE, NA, T >::Tail (
    const VEC\_TYPE & object_,
    R\_xlen\_t n_ ) [inline]
```

Definition at line 34 of file tail.h.

References [Rcpp::sugar::Tail< RTYPE, NA, T >::n](#), and [Rcpp::sugar::Tail< RTYPE, NA, T >::start](#).

## 6.771.4 Member Function Documentation

### 6.771.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >  
STORAGE Rcpp::sugar::Tail< RTYPE, NA, T >::operator[] (  
    R_xlen_t i) const [inline]
```

Definition at line 43 of file tail.h.

References Rcpp::sugar::Tail< RTYPE, NA, T >::start.

### 6.771.4.2 size()

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Tail< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 46 of file tail.h.

References Rcpp::sugar::Tail< RTYPE, NA, T >::n.

## 6.771.5 Member Data Documentation

### 6.771.5.1 n

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Tail< RTYPE, NA, T >::n [private]
```

Definition at line 50 of file tail.h.

Referenced by Rcpp::sugar::Tail< RTYPE, NA, T >::size(), and Rcpp::sugar::Tail< RTYPE, NA, T >::Tail().

### 6.771.5.2 object

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::sugar::Tail< RTYPE, NA, T >::object [private]
```

Definition at line 49 of file tail.h.

### 6.771.5.3 start

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::Tail< RTYPE, NA, T >::start [private]
```

Definition at line 50 of file tail.h.

Referenced by Rcpp::sugar::Tail< RTYPE, NA, T >::operator[](), and Rcpp::sugar::Tail< RTYPE, NA, T >::Tail().

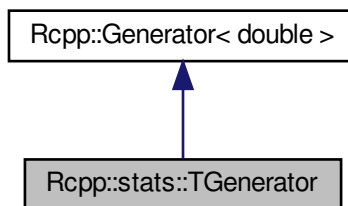
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/tail.h

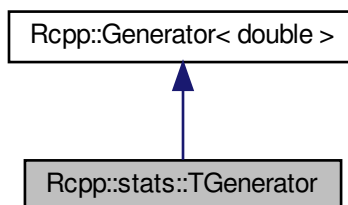
## 6.772 Rcpp::stats::TGenerator Class Reference

```
#include <rt.h>
```

Inheritance diagram for Rcpp::stats::TGenerator:



Collaboration diagram for Rcpp::stats::TGenerator:



## Public Member Functions

- [TGenerator](#) (double df\_)
- double [operator\(\)](#) () const

## Private Attributes

- double [df](#)
- double [df\\_2](#)

## Additional Inherited Members

### 6.772.1 Detailed Description

Definition at line 28 of file rt.h.

### 6.772.2 Constructor & Destructor Documentation

#### 6.772.2.1 TGenerator()

```
Rcpp::stats::TGenerator::TGenerator (  
    double df_ ) [inline]
```

Definition at line 31 of file rt.h.

### 6.772.3 Member Function Documentation

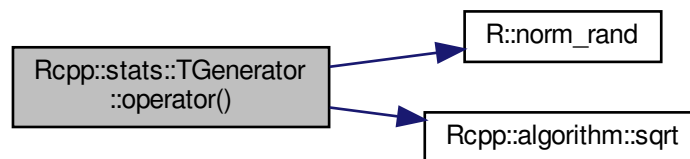
#### 6.772.3.1 operator()()

```
double Rcpp::stats::TGenerator::operator() ( ) const [inline]
```

Definition at line 33 of file rt.h.

References [df](#), [df\\_2](#), [R::norm\\_rand\(\)](#), and [Rcpp::algorithm::sqrt\(\)](#).

Here is the call graph for this function:



## 6.772.4 Member Data Documentation

### 6.772.4.1 df

```
double Rcpp::stats::TGenerator::df [private]
```

Definition at line 45 of file rt.h.

Referenced by operator(()).

### 6.772.4.2 df\_2

```
double Rcpp::stats::TGenerator::df_2 [private]
```

Definition at line 45 of file rt.h.

Referenced by operator(()).

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rt.h](#)

## 6.773 Rcpp::Timer Class Reference

```
#include <Timer.h>
```

### Public Member Functions

- [Timer](#) ()
- [Timer](#) ([nanotime\\_t](#) start\_time\_)
- void [step](#) (const std::string &name)
- [operator SEXP](#) () const
- [nanotime\\_t now](#) () const
- [nanotime\\_t origin](#) () const

### Static Public Member Functions

- static std::vector< [Timer](#) > [get\\_timers](#) (int n)



## Private Types

- typedef std::pair< std::string, [nanotime\\_t](#) > [Step](#)
- typedef std::vector< [Step](#) > [Steps](#)

## Private Attributes

- [Steps data](#)
- const [nanotime\\_t start\\_time](#)

### 6.773.1 Detailed Description

Definition at line 103 of file Timer.h.

### 6.773.2 Member Typedef Documentation

#### 6.773.2.1 Step

```
typedef std::pair<std::string,nanotime\_t> Rcpp::Timer::Step [private]
```

Definition at line 137 of file Timer.h.

#### 6.773.2.2 Steps

```
typedef std::vector<Step> Rcpp::Timer::Steps [private]
```

Definition at line 138 of file Timer.h.

### 6.773.3 Constructor & Destructor Documentation

#### 6.773.3.1 Timer() [1/2]

```
Rcpp::Timer::Timer ( ) [inline]
```

Definition at line 105 of file Timer.h.

Referenced by [get\\_timers\(\)](#).

### 6.773.3.2 Timer() [2/2]

```
Rcpp::Timer::Timer (
    nanotime_t start_time_ ) [inline]
```

Definition at line 106 of file Timer.h.

## 6.773.4 Member Function Documentation

### 6.773.4.1 get\_timers()

```
static std::vector<Timer> Rcpp::Timer::get_timers (
    int n ) [inline], [static]
```

Definition at line 124 of file Timer.h.

References Timer().

Here is the call graph for this function:



### 6.773.4.2 now()

```
nanotime_t Rcpp::Timer::now ( ) const [inline]
```

Definition at line 128 of file Timer.h.

Referenced by step().

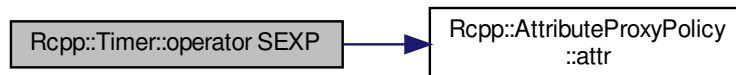
### 6.773.4.3 operator SEXP()

```
Rcpp::Timer::operator SEXP ( ) const [inline]
```

Definition at line 112 of file Timer.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), data, and start\_time.

Here is the call graph for this function:



### 6.773.4.4 origin()

```
nanotime_t Rcpp::Timer::origin ( ) const [inline]
```

Definition at line 132 of file Timer.h.

References start\_time.

### 6.773.4.5 step()

```
void Rcpp::Timer::step (
    const std::string & name ) [inline]
```

Definition at line 108 of file Timer.h.

References data, and now().

Here is the call graph for this function:



## 6.773.5 Member Data Documentation

### 6.773.5.1 data

`Steps` `Rcpp::Timer::data` [private]

Definition at line 140 of file `Timer.h`.

Referenced by operator `SEXP()`, and `step()`.

### 6.773.5.2 start\_time

`const nanotime_t` `Rcpp::Timer::start_time` [private]

Definition at line 141 of file `Timer.h`.

Referenced by operator `SEXP()`, and `origin()`.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/Benchmark/Timer.h](#)

## 6.774 Timer Class Reference

```
#include <Timer.h>
```

### Public Member Functions

- [Timer](#) ()
- void [Start](#) ()
- void [Stop](#) ()
- void [Reset](#) ()
- double [ElapsedTime](#) ()
- double [CumulativeTime](#) ()

### Private Member Functions

- double [getFractionalSeconds](#) (void)

## Private Attributes

- Function [sys\\_time](#)
- double [start\\_t](#)
- double [end\\_t](#)
- double [elapsed](#)
- double [cumul](#)

### 6.774.1 Detailed Description

#### Examples

[SugarPerformance/Timer.h](#), and [SugarPerformance/Timertest.cpp](#).

Definition at line 31 of file [Timer.h](#).

### 6.774.2 Constructor & Destructor Documentation

#### 6.774.2.1 Timer()

```
Timer::Timer ( ) [inline]
```

#### Examples

[SugarPerformance/Timer.h](#).

Definition at line 33 of file [Timer.h](#).

References [Reset\(\)](#).

Here is the call graph for this function:



### 6.774.3 Member Function Documentation

### 6.774.3.1 CumulativeTime()

```
double Timer::CumulativeTime ( ) [inline]
```

#### Examples

[SugarPerformance/Timer.h](#), and [SugarPerformance/Timertest.cpp](#).

Definition at line 42 of file `Timer.h`.

References `cumul`.

Referenced by `main()`.

### 6.774.3.2 ElapsedTime()

```
double Timer::ElapsedTime ( ) [inline]
```

#### Examples

[SugarPerformance/Timer.h](#), and [SugarPerformance/Timertest.cpp](#).

Definition at line 41 of file `Timer.h`.

References `elapsed`.

Referenced by `main()`.

### 6.774.3.3 getFractionalSeconds()

```
double Timer::getFractionalSeconds (
    void ) [inline], [private]
```

#### Examples

[SugarPerformance/Timer.h](#).

Definition at line 49 of file `Timer.h`.

References `sys_time`.

Referenced by `Start()`, and `Stop()`.

### 6.774.3.4 Reset()

```
void Timer::Reset ( ) [inline]
```

#### Examples

[SugarPerformance/Timer.h](#), and [SugarPerformance/Timertest.cpp](#).

Definition at line 40 of file Timer.h.

References `cumul`, `elapsed`, `end_t`, and `start_t`.

Referenced by `main()`, and `Timer()`.

### 6.774.3.5 Start()

```
void Timer::Start ( ) [inline]
```

#### Examples

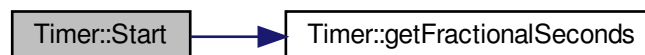
[SugarPerformance/Timer.h](#), and [SugarPerformance/Timertest.cpp](#).

Definition at line 34 of file Timer.h.

References `getFractionalSeconds()`, and `start_t`.

Referenced by `main()`.

Here is the call graph for this function:



### 6.774.3.6 Stop()

```
void Timer::Stop ( ) [inline]
```

#### Examples

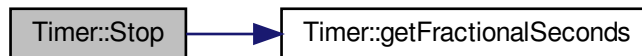
[SugarPerformance/Timer.h](#), and [SugarPerformance/Timertest.cpp](#).

Definition at line 35 of file `Timer.h`.

References `cumul`, `elapsed`, `end_t`, `getFractionalSeconds()`, and `start_t`.

Referenced by `main()`.

Here is the call graph for this function:



## 6.774.4 Member Data Documentation

### 6.774.4.1 cumul

```
double Timer::cumul [private]
```

#### Examples

[SugarPerformance/Timer.h](#).

Definition at line 47 of file `Timer.h`.

Referenced by `CumulativeTime()`, `Reset()`, and `Stop()`.



### 6.774.4.2 elapsed

```
double Timer::elapsed [private]
```

#### Examples

[SugarPerformance/Timer.h](#).

Definition at line 47 of file Timer.h.

Referenced by ElapsedTime(), Reset(), and Stop().

### 6.774.4.3 end\_t

```
double Timer::end_t [private]
```

#### Examples

[SugarPerformance/Timer.h](#).

Definition at line 47 of file Timer.h.

Referenced by Reset(), and Stop().

### 6.774.4.4 start\_t

```
double Timer::start_t [private]
```

#### Examples

[SugarPerformance/Timer.h](#).

Definition at line 47 of file Timer.h.

Referenced by Reset(), Start(), and Stop().

### 6.774.4.5 sys\_time

Function `Timer::sys_time` [private]

#### Examples

[SugarPerformance/Timer.h](#).

Definition at line 46 of file `Timer.h`.

Referenced by `getFractionalSeconds()`.

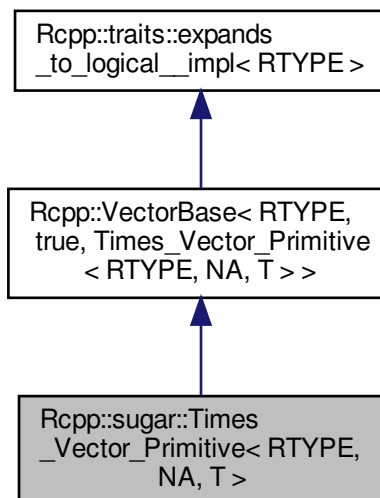
The documentation for this class was generated from the following file:

- [inst/examples/SugarPerformance/Timer.h](#)

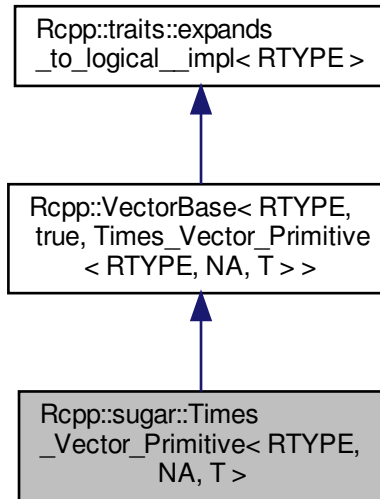
## 6.775 Rcpp::sugar::Times\_Vector\_Primitive< RTYPE, NA, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for `Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >`:



Collaboration diagram for Rcpp::sugar::Times\_Vector\_Primitive< RTYPE, NA, T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, NA, T > [VEC\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, NA, T >::type [EXT](#)

## Public Member Functions

- [Times\\_Vector\\_Primitive](#) (const [VEC\\_TYPE](#) &lhs\_, [STORAGE](#) rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [EXT](#) & [lhs](#)
- [STORAGE](#) [rhs](#)
- bool [rhs\\_na](#)

### 6.775.1 Detailed Description

```

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >
  
```

Definition at line 236 of file times.h.

## 6.775.2 Member Typedef Documentation

### 6.775.2.1 EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Times_Vector_Primitive< RTYPE,
NA, T >::EXT
```

Definition at line 241 of file times.h.

### 6.775.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >↔
::STORAGE
```

Definition at line 239 of file times.h.

### 6.775.2.3 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 238 of file times.h.

## 6.775.3 Constructor & Destructor Documentation

### 6.775.3.1 Times\_Vector\_Primitive()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::Times_Vector_Primitive (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 243 of file times.h.

## 6.775.4 Member Function Documentation

### 6.775.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 247 of file times.h.

References [Rcpp::sugar::Times\\_Vector\\_Primitive< RTYPE, NA, T >::lhs](#), [Rcpp::sugar::Times\\_Vector\\_Primitive< RTYPE, NA, T >::rhs](#), and [Rcpp::sugar::Times\\_Vector\\_Primitive< RTYPE, NA, T >::rhs\\_na](#).

### 6.775.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 253 of file times.h.

References [Rcpp::sugar::Times\\_Vector\\_Primitive< RTYPE, NA, T >::lhs](#).

## 6.775.5 Member Data Documentation

### 6.775.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::lhs [private]
```

Definition at line 256 of file times.h.

Referenced by [Rcpp::sugar::Times\\_Vector\\_Primitive< RTYPE, NA, T >::operator\[\]\(\)](#), [Rcpp::sugar::Times\\_Vector\\_Primitive< REALSXP, NA, T >::operator\[\]\(\)](#), [Rcpp::sugar::Times\\_Vector\\_Primitive< RTYPE, false, T >::operator\[\]\(\)](#), [Rcpp::sugar::Times\\_Vector\\_Primitive< REALSXP, false, T >::operator\[\]\(\)](#), [Rcpp::sugar::Times\\_Vector\\_Primitive< RTYPE, NA, T >::size\(\)](#), [Rcpp::sugar::Times\\_Vector\\_Primitive< REALSXP, NA, T >::size\(\)](#), [Rcpp::sugar::Times\\_Vector\\_Primitive< RTYPE, false, T >::size\(\)](#), and [Rcpp::sugar::Times\\_Vector\\_Primitive< REALSXP, false, T >::size\(\)](#).

### 6.775.5.2 rhs

```
template<int RTYPE, bool NA, typename T >  
STORAGE Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::rhs [private]
```

Definition at line 257 of file times.h.

Referenced by `Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::operator[]()`, `Rcpp::sugar::Times_Vector_↔ Primitive< REALSXP, NA, T >::operator[]()`, `Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::operator[]()`, and `Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::operator[]()`.

### 6.775.5.3 rhs\_na

```
template<int RTYPE, bool NA, typename T >  
bool Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::rhs_na [private]
```

Definition at line 258 of file times.h.

Referenced by `Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::operator[]()`, and `Rcpp::sugar::Times_↔ Vector_Primitive< RTYPE, false, T >::operator[]()`.

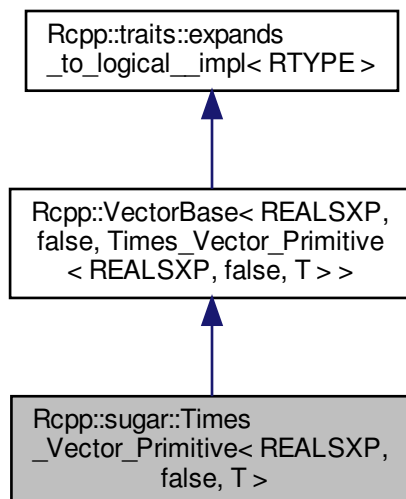
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/times.h](#)

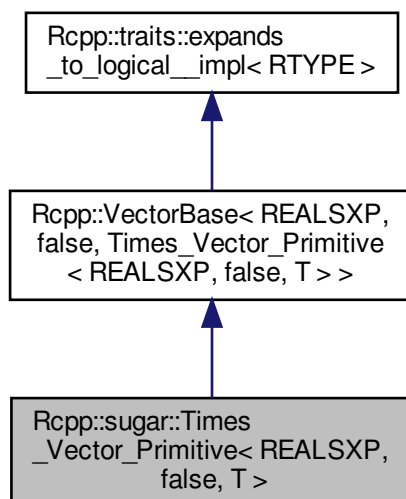
## 6.776 Rcpp::sugar::Times\_Vector\_Primitive< REALSXP, false, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Primitive< REALSXP, false, T >:



Collaboration diagram for Rcpp::sugar::Times\_Vector\_Primitive< REALSXP, false, T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, false, T > [VEC\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, T >::type [EXT](#)

## Public Member Functions

- [Times\\_Vector\\_Primitive](#) (const [VEC\\_TYPE](#) &lhs\_, double rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [EXT](#) & [lhs](#)
- double [rhs](#)

### 6.776.1 Detailed Description

```
template<typename T>
class Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >
```

Definition at line 309 of file times.h.

### 6.776.2 Member Typedef Documentation

#### 6.776.2.1 EXT

```
template<typename T >
typedef Rcpp::traits::Extractor<REALSXP, false, T>::type Rcpp::sugar::Times\_Vector\_Primitive<
REALSXP, false, T >::EXT
```

Definition at line 313 of file times.h.

#### 6.776.2.2 VEC\_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP, false, T> Rcpp::sugar::Times\_Vector\_Primitive< REALSXP, false, T
>::VEC_TYPE
```

Definition at line 312 of file times.h.



## 6.776.3 Constructor & Destructor Documentation

### 6.776.3.1 Times\_Vector\_Primitive()

```
template<typename T >
Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::Times_Vector_Primitive (
    const VEC_TYPE & lhs_,
    double rhs_ ) [inline]
```

Definition at line 315 of file times.h.

## 6.776.4 Member Function Documentation

### 6.776.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 318 of file times.h.

References [Rcpp::sugar::Times\\_Vector\\_Primitive< RTYPE, NA, T >::lhs](#), and [Rcpp::sugar::Times\\_Vector\\_Primitive< RTYPE, NA, T >::rhs](#).

### 6.776.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::size ( ) const [inline]
```

Definition at line 322 of file times.h.

References [Rcpp::sugar::Times\\_Vector\\_Primitive< RTYPE, NA, T >::lhs](#).

## 6.776.5 Member Data Documentation

**6.776.5.1 lhs**

```
template<typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::lhs [private]
```

Definition at line 325 of file times.h.

**6.776.5.2 rhs**

```
template<typename T >
double Rcpp::sugar::Times_Vector_Primitive< REALSXP, false, T >::rhs [private]
```

Definition at line 326 of file times.h.

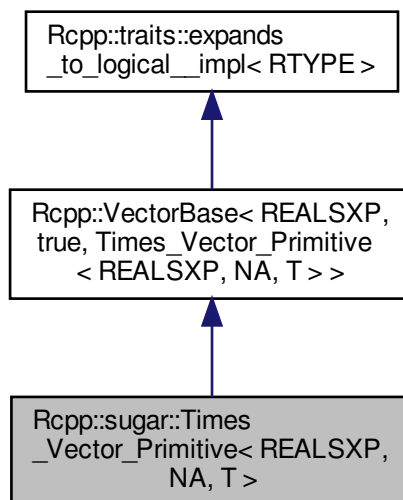
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/times.h](#)

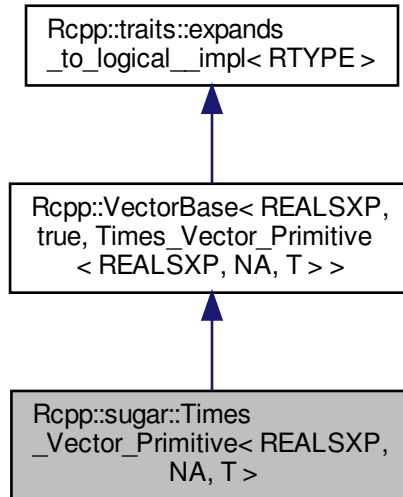
## 6.777 Rcpp::sugar::Times\_Vector\_Primitive< REALSXP, NA, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Primitive< REALSXP, NA, T >:



Collaboration diagram for Rcpp::sugar::Times\_Vector\_Primitive< REALSXP, NA, T >:



## Public Types

- typedef [Rcpp::VectorBase< REALSXP, NA, T >](#) [VEC\\_TYPE](#)
- typedef [Rcpp::traits::Extractor< REALSXP, NA, T >::type](#) [EXT](#)

## Public Member Functions

- [Times\\_Vector\\_Primitive](#) (const [VEC\\_TYPE](#) &lhs\_, double rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [EXT](#) & lhs
- double rhs

### 6.777.1 Detailed Description

```

template<bool NA, typename T>
class Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >

```

Definition at line 262 of file times.h.

## 6.777.2 Member Typedef Documentation

### 6.777.2.1 EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP, NA, T>::type Rcpp::sugar::Times_Vector_Primitive< REALSXP,
NA, T >::EXT
```

Definition at line 266 of file times.h.

### 6.777.2.2 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >↔
::VEC_TYPE
```

Definition at line 265 of file times.h.

## 6.777.3 Constructor & Destructor Documentation

### 6.777.3.1 Times\_Vector\_Primitive()

```
template<bool NA, typename T >
Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::Times_Vector_Primitive (
    const VEC_TYPE & lhs_,
    double rhs_ ) [inline]
```

Definition at line 268 of file times.h.

## 6.777.4 Member Function Documentation

### 6.777.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::operator[] (
    R_xlen_t i) const [inline]
```

Definition at line 272 of file times.h.

References Rcpp::sugar::Times\_Vector\_Primitive< RTYPE, NA, T >::lhs, and Rcpp::sugar::Times\_Vector\_Primitive< RTYPE, NA, T >::rhs.

### 6.777.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::size ( ) const [inline]
```

Definition at line 276 of file times.h.

References Rcpp::sugar::Times\_Vector\_Primitive< RTYPE, NA, T >::lhs.

## 6.777.5 Member Data Documentation

### 6.777.5.1 lhs

```
template<bool NA, typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::lhs [private]
```

Definition at line 279 of file times.h.

### 6.777.5.2 rhs

```
template<bool NA, typename T >
double Rcpp::sugar::Times_Vector_Primitive< REALSXP, NA, T >::rhs [private]
```

Definition at line 280 of file times.h.

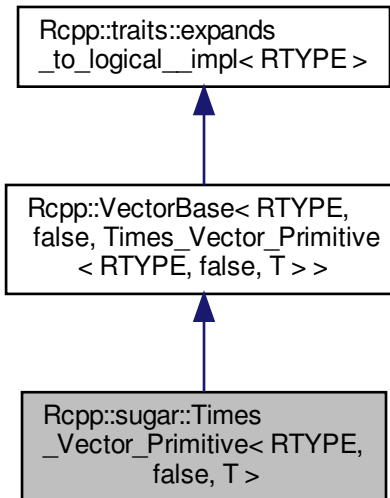
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

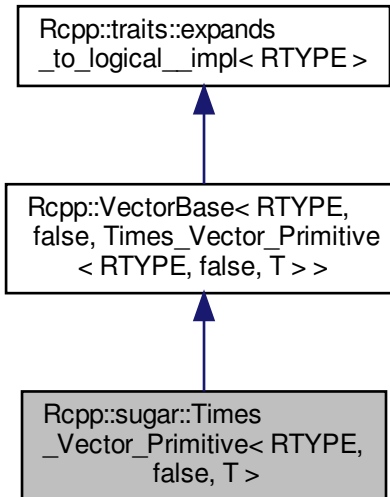
## 6.778 Rcpp::sugar::Times\_Vector\_Primitive< RTYPE, false, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Primitive< RTYPE, false, T >:



Collaboration diagram for Rcpp::sugar::Times\_Vector\_Primitive< RTYPE, false, T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, false, T > [VEC\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, T >::type [EXT](#)

## Public Member Functions

- [Times\\_Vector\\_Primitive](#) (const [VEC\\_TYPE](#) &lhs\_, [STORAGE](#) rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [EXT](#) & [lhs](#)
- [STORAGE](#) [rhs](#)
- bool [rhs\\_na](#)

### 6.778.1 Detailed Description

```

template<int RTYPE, typename T>
class Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >
  
```

Definition at line 285 of file times.h.

## 6.778.2 Member Typedef Documentation

### 6.778.2.1 EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor< RTYPE, false, T>::type Rcpp::sugar::Times_Vector_Primitive<
RTYPE, false, T >::EXT
```

Definition at line 290 of file times.h.

### 6.778.2.2 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T
>::STORAGE
```

Definition at line 288 of file times.h.

### 6.778.2.3 VEC\_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >↔
::VEC_TYPE
```

Definition at line 287 of file times.h.

## 6.778.3 Constructor & Destructor Documentation

### 6.778.3.1 Times\_Vector\_Primitive()

```
template<int RTYPE, typename T >
Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::Times_Vector_Primitive (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 292 of file times.h.



## 6.778.4 Member Function Documentation

### 6.778.4.1 operator[]()

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::operator[] (  
    R_xlen_t i ) const [inline]
```

Definition at line 295 of file times.h.

References `Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::lhs`, `Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::rhs`, and `Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::rhs_na`.

### 6.778.4.2 size()

```
template<int RTYPE, typename T >  
R_xlen_t Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::size ( ) const [inline]
```

Definition at line 299 of file times.h.

References `Rcpp::sugar::Times_Vector_Primitive< RTYPE, NA, T >::lhs`.

## 6.778.5 Member Data Documentation

### 6.778.5.1 lhs

```
template<int RTYPE, typename T >  
const EXT& Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::lhs [private]
```

Definition at line 302 of file times.h.

### 6.778.5.2 rhs

```
template<int RTYPE, typename T >  
STORAGE Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::rhs [private]
```

Definition at line 303 of file times.h.

### 6.778.5.3 rhs\_na

```
template<int RTYPE, typename T >
bool Rcpp::sugar::Times_Vector_Primitive< RTYPE, false, T >::rhs_na [private]
```

Definition at line 304 of file times.h.

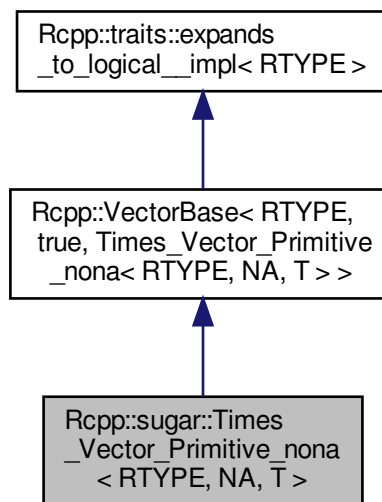
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/times.h](#)

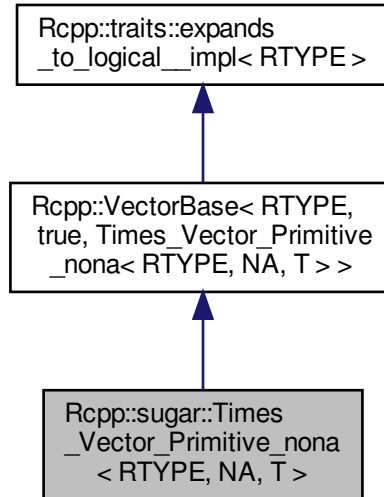
## 6.779 Rcpp::sugar::Times\_Vector\_Primitive\_nona< RTYPE, NA, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Primitive\_nona< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Times\_Vector\_Primitive\_nona< RTYPE, NA, T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, NA, T > [VEC\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, NA, T >::type [EXT](#)

## Public Member Functions

- [Times\\_Vector\\_Primitive\\_nona](#) (const [VEC\\_TYPE](#) &lhs\_, [STORAGE](#) rhs\_)
- [STORAGE](#) operator[] (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [EXT](#) & lhs
- [STORAGE](#) rhs

### 6.779.1 Detailed Description

```

template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >

```

Definition at line 336 of file times.h.

## 6.779.2 Member Typedef Documentation

### 6.779.2.1 EXT

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::Extractor< RTYPE, NA, T>::type Rcpp::sugar::Times_Vector_Primitive_nona<
RTYPE, NA, T >::EXT
```

Definition at line 340 of file times.h.

### 6.779.2.2 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T
>::STORAGE
```

Definition at line 339 of file times.h.

### 6.779.2.3 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >↔
::VEC_TYPE
```

Definition at line 338 of file times.h.

## 6.779.3 Constructor & Destructor Documentation

### 6.779.3.1 Times\_Vector\_Primitive\_nona()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::Times_Vector_Primitive_nona (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 342 of file times.h.

## 6.779.4 Member Function Documentation

### 6.779.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 346 of file times.h.

References Rcpp::sugar::Times\_Vector\_Primitive\_nona< RTYPE, NA, T >::lhs, and Rcpp::sugar::Times\_Vector\_Primitive\_nona< RTYPE, NA, T >::rhs.

### 6.779.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 351 of file times.h.

References Rcpp::sugar::Times\_Vector\_Primitive\_nona< RTYPE, NA, T >::lhs.

## 6.779.5 Member Data Documentation

### 6.779.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs [private]
```

Definition at line 354 of file times.h.

Referenced by Rcpp::sugar::Times\_Vector\_Primitive\_nona< RTYPE, NA, T >::operator[](), Rcpp::sugar::Times\_Vector\_Primitive\_nona< REALSXP, NA, T >::operator[](), Rcpp::sugar::Times\_Vector\_Primitive\_nona< RTYPE, false, T >::operator[](), Rcpp::sugar::Times\_Vector\_Primitive\_nona< REALSXP, false, T >::operator[](), Rcpp::sugar::Times\_Vector\_Primitive\_nona< RTYPE, NA, T >::size(), Rcpp::sugar::Times\_Vector\_Primitive\_nona< REALSXP, NA, T >::size(), Rcpp::sugar::Times\_Vector\_Primitive\_nona< RTYPE, false, T >::size(), and Rcpp::sugar::Times\_Vector\_Primitive\_nona< REALSXP, false, T >::size().

### 6.779.5.2 rhs

```
template<int RTYPE, bool NA, typename T >
STORAGE Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::rhs [private]
```

Definition at line 355 of file times.h.

Referenced by `Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::operator[]()`, `Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::operator[]()`, `Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::operator[]()`, and `Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::operator[]()`.

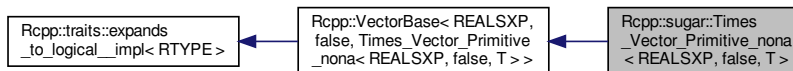
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/times.h](#)

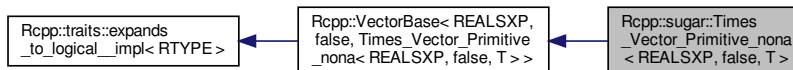
## 6.780 Rcpp::sugar::Times\_Vector\_Primitive\_nona< REALSXP, false, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for `Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >`:



Collaboration diagram for `Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >`:



### Public Types

- typedef `Rcpp::VectorBase< REALSXP, false, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< REALSXP, false, T >` `EXT`

## Public Member Functions

- [Times\\_Vector\\_Primitive\\_nona](#) (const [VEC\\_TYPE](#) &lhs\_, double rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [EXT](#) & lhs
- double [rhs](#)

### 6.780.1 Detailed Description

```
template<typename T>
class Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >
```

Definition at line 405 of file times.h.

### 6.780.2 Member Typedef Documentation

#### 6.780.2.1 EXT

```
template<typename T >
typedef Rcpp::traits::Extractor<REALSXP, false, T>::type Rcpp::sugar::Times_Vector_Primitive_nona<
REALSXP, false, T >::EXT
```

Definition at line 409 of file times.h.

#### 6.780.2.2 VEC\_TYPE

```
template<typename T >
typedef Rcpp::VectorBase<REALSXP, false, T> Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP,
false, T >::VEC_TYPE
```

Definition at line 408 of file times.h.

### 6.780.3 Constructor & Destructor Documentation

### 6.780.3.1 Times\_Vector\_Primitive\_nona()

```
template<typename T >
Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::Times_Vector_Primitive_nona (
    const VEC_TYPE & lhs_,
    double rhs_ ) [inline]
```

Definition at line 411 of file times.h.

## 6.780.4 Member Function Documentation

### 6.780.4.1 operator[]()

```
template<typename T >
double Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 414 of file times.h.

References `Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs`, and `Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::rhs`.

### 6.780.4.2 size()

```
template<typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::size ( ) const [inline]
```

Definition at line 418 of file times.h.

References `Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs`.

## 6.780.5 Member Data Documentation

### 6.780.5.1 lhs

```
template<typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::lhs [private]
```

Definition at line 421 of file times.h.



### 6.780.5.2 rhs

```
template<typename T >  
double Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, false, T >::rhs [private]
```

Definition at line 422 of file times.h.

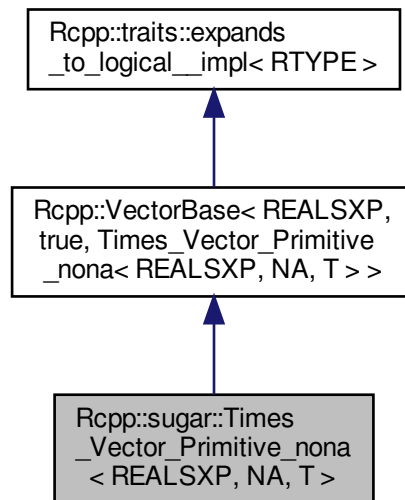
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

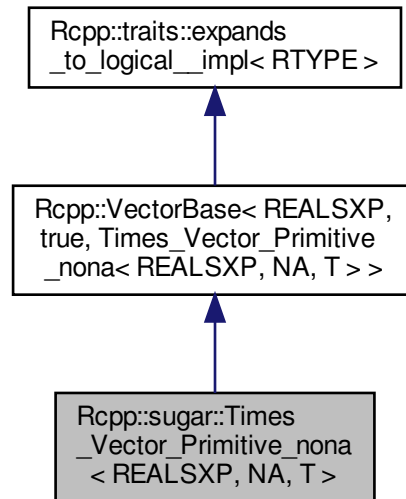
## 6.781 Rcpp::sugar::Times\_Vector\_Primitive\_nona< REALSXP, NA, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Primitive\_nona< REALSXP, NA, T >:



Collaboration diagram for `Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >`:



## Public Types

- typedef `Rcpp::VectorBase< REALSXP, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< REALSXP, NA, T >::type` `EXT`

## Public Member Functions

- `Times_Vector_Primitive_nona` (const `VEC_TYPE` &lhs\_, double rhs\_)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const `EXT` & lhs
- double rhs

### 6.781.1 Detailed Description

```

template<bool NA, typename T>
class Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >
  
```

Definition at line 360 of file times.h.

## 6.781.2 Member Typedef Documentation

### 6.781.2.1 EXT

```
template<bool NA, typename T >
typedef Rcpp::traits::Extractor<REALSXP,NA,T>::type Rcpp::sugar::Times_Vector_Primitive_nona<
REALSXP, NA, T >::EXT
```

Definition at line 364 of file times.h.

### 6.781.2.2 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<REALSXP,NA,T> Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T
>::VEC_TYPE
```

Definition at line 363 of file times.h.

## 6.781.3 Constructor & Destructor Documentation

### 6.781.3.1 Times\_Vector\_Primitive\_nona()

```
template<bool NA, typename T >
Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::Times_Vector_Primitive_nona (
    const VEC_TYPE & lhs_,
    double rhs_ ) [inline]
```

Definition at line 366 of file times.h.

## 6.781.4 Member Function Documentation

### 6.781.4.1 operator[]()

```
template<bool NA, typename T >
double Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::operator[] (
    R_xlen_t i) const [inline]
```

Definition at line 370 of file times.h.

References `Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs`, and `Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::rhs`.

### 6.781.4.2 size()

```
template<bool NA, typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::size ( ) const [inline]
```

Definition at line 374 of file times.h.

References `Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs`.

## 6.781.5 Member Data Documentation

### 6.781.5.1 lhs

```
template<bool NA, typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::lhs [private]
```

Definition at line 377 of file times.h.

### 6.781.5.2 rhs

```
template<bool NA, typename T >
double Rcpp::sugar::Times_Vector_Primitive_nona< REALSXP, NA, T >::rhs [private]
```

Definition at line 378 of file times.h.

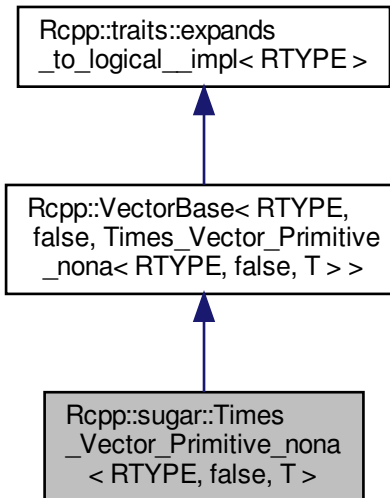
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/operators/times.h`

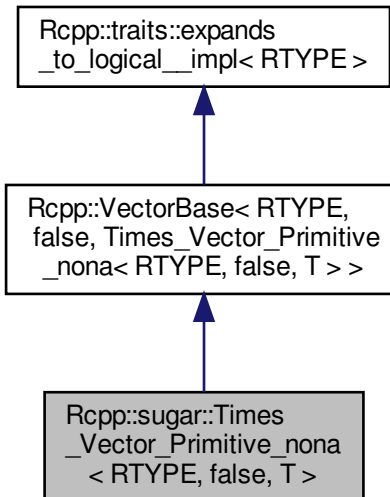
## 6.782 Rcpp::sugar::Times\_Vector\_Primitive\_nona< RTYPE, false, T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Primitive\_nona< RTYPE, false, T >:



Collaboration diagram for `Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >`:



## Public Types

- typedef `Rcpp::VectorBase< RTYPE, false, T >` `VEC_TYPE`
- typedef `traits::storage_type< RTYPE >::type` `STORAGE`
- typedef `Rcpp::traits::Extractor< RTYPE, false, T >::type` `EXT`

## Public Member Functions

- `Times_Vector_Primitive_nona` (const `VEC_TYPE` &lhs\_, `STORAGE` rhs\_)
- `STORAGE operator[]` (`R_xlen_t` i) const
- `R_xlen_t size` () const

## Private Attributes

- const `EXT` & lhs
- `STORAGE` rhs

### 6.782.1 Detailed Description

```

template<int RTYPE, typename T>
class Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >

```

Definition at line 383 of file times.h.

## 6.782.2 Member Typedef Documentation

### 6.782.2.1 EXT

```
template<int RTYPE, typename T >
typedef Rcpp::traits::Extractor< RTYPE, false, T>::type Rcpp::sugar::Times_Vector_Primitive_nona<
RTYPE, false, T >::EXT
```

Definition at line 388 of file times.h.

### 6.782.2.2 STORAGE

```
template<int RTYPE, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false,
T >::STORAGE
```

Definition at line 386 of file times.h.

### 6.782.2.3 VEC\_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T
>::VEC_TYPE
```

Definition at line 385 of file times.h.

## 6.782.3 Constructor & Destructor Documentation

### 6.782.3.1 Times\_Vector\_Primitive\_nona()

```
template<int RTYPE, typename T >
Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::Times_Vector_Primitive_nona (
    const VEC_TYPE & lhs_,
    STORAGE rhs_ ) [inline]
```

Definition at line 390 of file times.h.

## 6.782.4 Member Function Documentation

### 6.782.4.1 operator[]()

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 393 of file times.h.

References `Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs`, and `Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::rhs`.

### 6.782.4.2 size()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::size ( ) const [inline]
```

Definition at line 397 of file times.h.

References `Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, NA, T >::lhs`.

## 6.782.5 Member Data Documentation

### 6.782.5.1 lhs

```
template<int RTYPE, typename T >
const EXT& Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::lhs [private]
```

Definition at line 400 of file times.h.

### 6.782.5.2 rhs

```
template<int RTYPE, typename T >
STORAGE Rcpp::sugar::Times_Vector_Primitive_nona< RTYPE, false, T >::rhs [private]
```

Definition at line 401 of file times.h.

The documentation for this class was generated from the following file:

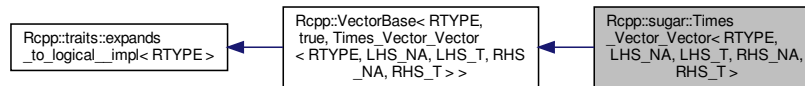
- `inst/include/Rcpp/sugar/operators/times.h`



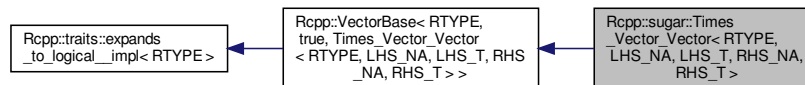
## 6.783 Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, RHS\_NA, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Times\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

### 6.783.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>  
class Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 29 of file times.h.

### 6.783.2 Member Typedef Documentation

#### 6.783.2.1 LHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<  
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 35 of file times.h.

#### 6.783.2.2 LHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::VectorBase<RTYPE, LHS_NA, LHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA,  
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 32 of file times.h.

#### 6.783.2.3 RHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >  
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<  
RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 36 of file times.h.

### 6.783.2.4 RHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 33 of file times.h.

### 6.783.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 31 of file times.h.

## 6.783.3 Constructor & Destructor Documentation

### 6.783.3.1 Times\_Vector\_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Times_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 38 of file times.h.

## 6.783.4 Member Function Documentation

### 6.783.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 42 of file times.h.

References Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.783.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 49 of file times.h.

References Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs.

## 6.783.5 Member Data Documentation

### 6.783.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 52 of file times.h.

Referenced by Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Times\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::← Times\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Times\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >::operator[](), Rcpp::sugar::Times\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS← \_T >::operator[](), Rcpp::sugar::Times\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >::operator[](), Rcpp← :sugar::Times\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >::operator[](), Rcpp::sugar::Times\_Vector\_← Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::Times\_Vector\_Vector< REALSXP, LHS← \_NA, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::Times\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS← \_T >::size(), Rcpp::sugar::Times\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar← :Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >::size(), Rcpp::sugar::Times\_Vector\_Vector< RE← ALSXP, LHS\_NA, LHS\_T, false, RHS\_T >::size(), Rcpp::sugar::Times\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >::size(), and Rcpp::sugar::Times\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >::size().

### 6.783.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs [private]
```

Definition at line 53 of file times.h.

Referenced by Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Times\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::← Times\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Times\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >::operator[](), Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >::operator[](), Rcpp::sugar::Times\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >::operator[](), Rcpp::sugar::Times\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >::operator[](), and Rcpp← :sugar::Times\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >::operator[]().

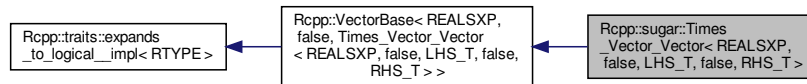
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

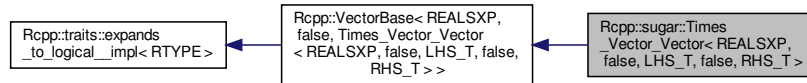
## 6.784 Rcpp::sugar::Times\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Times\_Vector\_Vector< REALSXP, false, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< REALSXP, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Times\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) & [lhs](#)
- const [RHS\\_EXT](#) & [rhs](#)

## 6.784.1 Detailed Description

```
template<typename LHS_T, typename RHS_T>  
class Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >
```

Definition at line 211 of file times.h.

## 6.784.2 Member Typedef Documentation

### 6.784.2.1 LHS\_EXT

```
template<typename LHS_T , typename RHS_T >  
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<  
REALSXP, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 217 of file times.h.

### 6.784.2.2 LHS\_TYPE

```
template<typename LHS_T , typename RHS_T >  
typedef Rcpp::VectorBase<REALSXP, false, LHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, false,  
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 214 of file times.h.

### 6.784.2.3 RHS\_EXT

```
template<typename LHS_T , typename RHS_T >  
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<  
REALSXP, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 218 of file times.h.

### 6.784.2.4 RHS\_TYPE

```
template<typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, false, RHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, false,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 215 of file times.h.

## 6.784.3 Constructor & Destructor Documentation

### 6.784.3.1 Times\_Vector\_Vector()

```
template<typename LHS_T , typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::Times_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 220 of file times.h.

## 6.784.4 Member Function Documentation

### 6.784.4.1 operator[]()

```
template<typename LHS_T , typename RHS_T >
double Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 223 of file times.h.

References [Rcpp::sugar::Times\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#), and [Rcpp::sugar::Times\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::rhs](#).

### 6.784.4.2 size()

```
template<typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 227 of file times.h.

References [Rcpp::sugar::Times\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#).

## 6.784.5 Member Data Documentation

### 6.784.5.1 lhs

```
template<typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::lhs [private]
```

Definition at line 230 of file times.h.

### 6.784.5.2 rhs

```
template<typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, false, RHS_T >::rhs [private]
```

Definition at line 231 of file times.h.

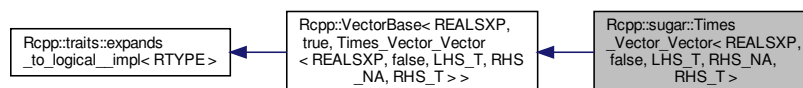
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/times.h](#)

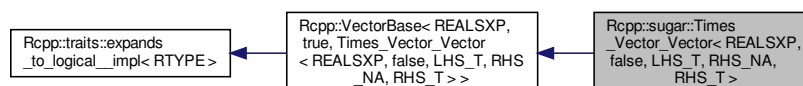
## 6.785 Rcpp::sugar::Times\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Times\_Vector\_Vector< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >:





## Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< REALSXP, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, RHS\_NA, RHS\_T >::type [RHS\\_EXT](#)

## Public Member Functions

- [Times\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & [lhs](#)
- const [RHS\\_EXT](#) & [rhs](#)

### 6.785.1 Detailed Description

```
template<typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >
```

Definition at line 110 of file times.h.

### 6.785.2 Member Typedef Documentation

#### 6.785.2.1 LHS\_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, LHS_T>::type Rcpp::sugar::Times\_Vector\_Vector<
REALSXP, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 116 of file times.h.

### 6.785.2.2 LHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, false, LHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 113 of file times.h.

### 6.785.2.3 RHS\_EXT

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<
REALSXP, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 117 of file times.h.

### 6.785.2.4 RHS\_TYPE

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, RHS_NA, RHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 114 of file times.h.

## 6.785.3 Constructor & Destructor Documentation

### 6.785.3.1 Times\_Vector\_Vector()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::Times_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 119 of file times.h.

### 6.785.4 Member Function Documentation

### 6.785.4.1 operator[]()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 122 of file times.h.

References Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.785.4.2 size()

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 126 of file times.h.

References Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs.

## 6.785.5 Member Data Documentation

### 6.785.5.1 lhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 129 of file times.h.

### 6.785.5.2 rhs

```
template<typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, false, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 130 of file times.h.

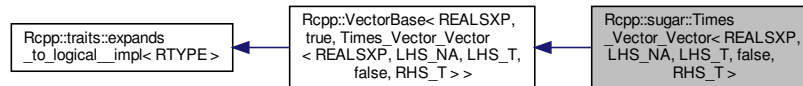
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

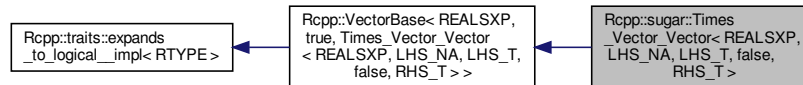
## 6.786 Rcpp::sugar::Times\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Times\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< REALSXP, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, false, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Times\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

## 6.786.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >
```

Definition at line 162 of file times.h.

## 6.786.2 Member Typedef Documentation

### 6.786.2.1 LHS\_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 168 of file times.h.

### 6.786.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, LHS_NA, LHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA,
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 165 of file times.h.

### 6.786.2.3 RHS\_EXT

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, false, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<
REALSXP, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 169 of file times.h.

### 6.786.2.4 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<REALSXP, false, RHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 166 of file times.h.

## 6.786.3 Constructor & Destructor Documentation

### 6.786.3.1 Times\_Vector\_Vector()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::Times_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 171 of file times.h.

## 6.786.4 Member Function Documentation

### 6.786.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
double Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 174 of file times.h.

References [Rcpp::sugar::Times\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#), and [Rcpp::sugar::Times\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::rhs](#).

### 6.786.4.2 size()

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 178 of file times.h.

References [Rcpp::sugar::Times\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >::lhs](#).

## 6.786.5 Member Data Documentation

### 6.786.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::lhs
[private]
```

Definition at line 181 of file times.h.

### 6.786.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, false, RHS_T >::rhs
[private]
```

Definition at line 182 of file times.h.

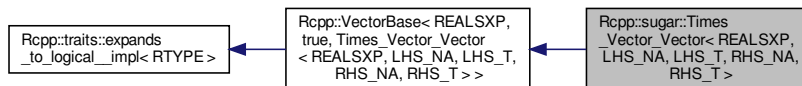
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/times.h

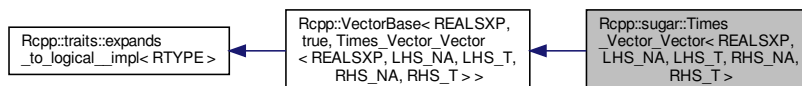
## 6.787 Rcpp::sugar::Times\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Times\_Vector\_Vector< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< REALSXP, LHS\_NA, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< REALSXP, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, LHS\_NA, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< REALSXP, RHS\_NA, RHS\_T >::type [RHS\\_EXT](#)

## Public Member Functions

- [Times\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [LHS\\_EXT](#) & lhs
- const [RHS\\_EXT](#) & rhs

### 6.787.1 Detailed Description

```
template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 57 of file times.h.

### 6.787.2 Member Typedef Documentation

#### 6.787.2.1 LHS\_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, LHS_NA, LHS_T>::type Rcpp::sugar::Times\_Vector\_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 63 of file times.h.



### 6.787.2.2 LHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,LHS_NA,LHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA,
LHS_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 60 of file times.h.

### 6.787.2.3 RHS\_EXT

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor<REALSXP, RHS_NA, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<
REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 64 of file times.h.

### 6.787.2.4 RHS\_TYPE

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<REALSXP,RHS_NA,RHS_T> Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 61 of file times.h.

## 6.787.3 Constructor & Destructor Documentation

### 6.787.3.1 Times\_Vector\_Vector()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::Times_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 66 of file times.h.

### 6.787.4 Member Function Documentation

### 6.787.4.1 operator[]()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
double Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 70 of file times.h.

References `Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`, and `Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs`.

### 6.787.4.2 size()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::size ( )
const [inline]
```

Definition at line 74 of file times.h.

References `Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

## 6.787.5 Member Data Documentation

### 6.787.5.1 lhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs
[private]
```

Definition at line 77 of file times.h.

### 6.787.5.2 rhs

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< REALSXP, LHS_NA, LHS_T, RHS_NA, RHS_T >::rhs
[private]
```

Definition at line 78 of file times.h.

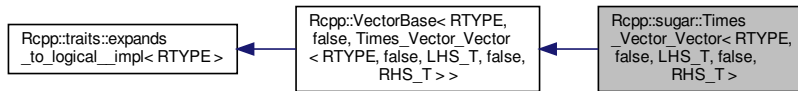
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/operators/times.h`

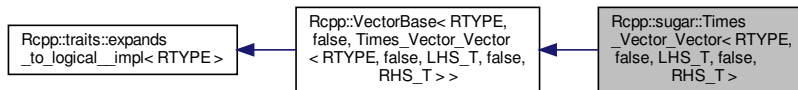
## 6.788 Rcpp::sugar::Times\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Times\_Vector\_Vector< RTYPE, false, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, false, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Times\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

## 6.788.1 Detailed Description

```
template<int RTYPE, typename LHS_T, typename RHS_T>
class Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >
```

Definition at line 187 of file times.h.

## 6.788.2 Member Typedef Documentation

### 6.788.2.1 LHS\_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 193 of file times.h.

### 6.788.2.2 LHS\_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T,
false, RHS_T >::LHS_TYPE
```

Definition at line 190 of file times.h.

### 6.788.2.3 RHS\_EXT

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<
RTYPE, false, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 194 of file times.h.

### 6.788.2.4 RHS\_TYPE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T,
false, RHS_T >::RHS_TYPE
```

Definition at line 191 of file times.h.

### 6.788.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T,
false, RHS_T >::STORAGE
```

Definition at line 189 of file times.h.

## 6.788.3 Constructor & Destructor Documentation

### 6.788.3.1 Times\_Vector\_Vector()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::Times_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 196 of file times.h.

## 6.788.4 Member Function Documentation

### 6.788.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 199 of file times.h.

References Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.788.4.2 size()

```
template<int RTYPE, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 203 of file times.h.

References `Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

## 6.788.5 Member Data Documentation

### 6.788.5.1 lhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::lhs [private]
```

Definition at line 206 of file times.h.

### 6.788.5.2 rhs

```
template<int RTYPE, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, false, RHS_T >::rhs [private]
```

Definition at line 207 of file times.h.

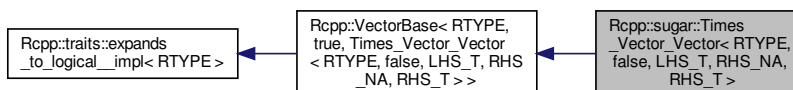
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/times.h](#)

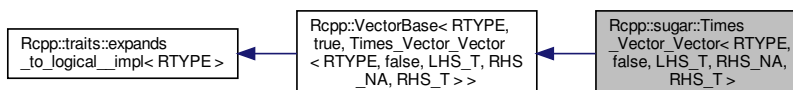
## 6.789 Rcpp::sugar::Times\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Times\_Vector\_Vector< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >:



### Public Types

- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, false, LHS\_T > [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, false, LHS\_T >::type [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor](#)< RTYPE, RHS\_NA, RHS\_T >::type [RHS\\_EXT](#)

### Public Member Functions

- [Times\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

## 6.789.1 Detailed Description

```
template<int RTYPE, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >
```

Definition at line 84 of file times.h.

## 6.789.2 Member Typedef Documentation

### 6.789.2.1 LHS\_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, false, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::LHS_EXT
```

Definition at line 90 of file times.h.

### 6.789.2.2 LHS\_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,LHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS↔
_T, RHS_NA, RHS_T >::LHS_TYPE
```

Definition at line 87 of file times.h.

### 6.789.2.3 RHS\_EXT

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::Extractor< RTYPE, RHS_NA, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<
RTYPE, false, LHS_T, RHS_NA, RHS_T >::RHS_EXT
```

Definition at line 91 of file times.h.



### 6.789.2.4 RHS\_TYPE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,RHS_NA,RHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, false,
LHS_T, RHS_NA, RHS_T >::RHS_TYPE
```

Definition at line 88 of file times.h.

### 6.789.2.5 STORAGE

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T,
RHS_NA, RHS_T >::STORAGE
```

Definition at line 86 of file times.h.

## 6.789.3 Constructor & Destructor Documentation

### 6.789.3.1 Times\_Vector\_Vector()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::Times_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 93 of file times.h.

## 6.789.4 Member Function Documentation

### 6.789.4.1 operator[]()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
STORAGE Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 96 of file times.h.

References Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.789.4.2 size()

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::size ( ) const
[inline]
```

Definition at line 102 of file times.h.

References `Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

## 6.789.5 Member Data Documentation

### 6.789.5.1 lhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::lhs [private]
```

Definition at line 105 of file times.h.

### 6.789.5.2 rhs

```
template<int RTYPE, typename LHS_T , bool RHS_NA, typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, false, LHS_T, RHS_NA, RHS_T >::rhs [private]
```

Definition at line 106 of file times.h.

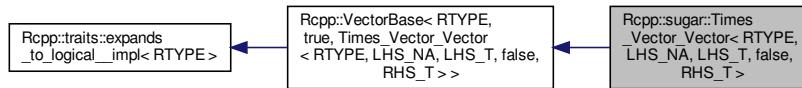
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/times.h](#)

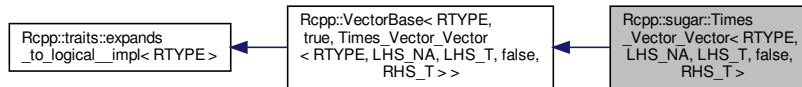
## 6.790 Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T > Class Template Reference

```
#include <times.h>
```

Inheritance diagram for Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >:



Collaboration diagram for Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >:



### Public Types

- typedef [traits::storage\\_type< RTYPE >::type](#) [STORAGE](#)
- typedef [Rcpp::VectorBase< RTYPE, LHS\\_NA, LHS\\_T >](#) [LHS\\_TYPE](#)
- typedef [Rcpp::VectorBase< RTYPE, false, RHS\\_T >](#) [RHS\\_TYPE](#)
- typedef [Rcpp::traits::Extractor< RTYPE, LHS\\_NA, LHS\\_T >::type](#) [LHS\\_EXT](#)
- typedef [Rcpp::traits::Extractor< RTYPE, false, RHS\\_T >::type](#) [RHS\\_EXT](#)

### Public Member Functions

- [Times\\_Vector\\_Vector](#) (const [LHS\\_TYPE](#) &lhs\_, const [RHS\\_TYPE](#) &rhs\_)
- [STORAGE operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

### Private Attributes

- const [LHS\\_EXT](#) &lhs
- const [RHS\\_EXT](#) &rhs

## 6.790.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, typename RHS_T>  
class Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >
```

Definition at line 136 of file times.h.

## 6.790.2 Member Typedef Documentation

### 6.790.2.1 LHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef Rcpp::traits::Extractor< RTYPE, LHS_NA, LHS_T>::type Rcpp::sugar::Times_Vector_Vector<  
RTYPE, LHS_NA, LHS_T, false, RHS_T >::LHS_EXT
```

Definition at line 142 of file times.h.

### 6.790.2.2 LHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef Rcpp::VectorBase<RTYPE, LHS_NA, LHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA,  
LHS_T, false, RHS_T >::LHS_TYPE
```

Definition at line 139 of file times.h.

### 6.790.2.3 RHS\_EXT

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >  
typedef Rcpp::traits::Extractor< RTYPE, false, RHS_T>::type Rcpp::sugar::Times_Vector_Vector<  
RTYPE, LHS_NA, LHS_T, false, RHS_T >::RHS_EXT
```

Definition at line 143 of file times.h.

### 6.790.2.4 RHS\_TYPE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef Rcpp::VectorBase<RTYPE,false,RHS_T> Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA,
LHS_T, false, RHS_T >::RHS_TYPE
```

Definition at line 140 of file times.h.

### 6.790.2.5 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T,
false, RHS_T >::STORAGE
```

Definition at line 138 of file times.h.

## 6.790.3 Constructor & Destructor Documentation

### 6.790.3.1 Times\_Vector\_Vector()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::Times_Vector_Vector (
    const LHS_TYPE & lhs_,
    const RHS_TYPE & rhs_ ) [inline]
```

Definition at line 145 of file times.h.

## 6.790.4 Member Function Documentation

### 6.790.4.1 operator[]()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
STORAGE Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 148 of file times.h.

References Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::lhs, and Rcpp::sugar::Times\_Vector\_Vector< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::rhs.

### 6.790.4.2 size()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
R_xlen_t Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::size ( ) const
[inline]
```

Definition at line 154 of file times.h.

References `Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::lhs`.

## 6.790.5 Member Data Documentation

### 6.790.5.1 lhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const LHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::lhs [private]
```

Definition at line 157 of file times.h.

### 6.790.5.2 rhs

```
template<int RTYPE, bool LHS_NA, typename LHS_T , typename RHS_T >
const RHS_EXT& Rcpp::sugar::Times_Vector_Vector< RTYPE, LHS_NA, LHS_T, false, RHS_T >::rhs [private]
```

Definition at line 158 of file times.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/operators/times.h](#)

## 6.791 Rcpp::ttinfo Struct Reference

### Public Attributes

- [int\\_fast32\\_t tt\\_gmtoff](#)
- [int tt\\_isdst](#)
- [int tt\\_abbrind](#)
- [int tt\\_tisstd](#)
- [int tt\\_tisgmt](#)

## 6.791.1 Detailed Description

Definition at line 380 of file date.cpp.

## 6.791.2 Member Data Documentation

### 6.791.2.1 tt\_abbrind

```
int Rcpp::ttinfo::tt_abbrind
```

Definition at line 383 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

### 6.791.2.2 tt\_gmtoff

```
int_fast32_t Rcpp::ttinfo::tt_gmtoff
```

Definition at line 381 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

### 6.791.2.3 tt\_isdst

```
int Rcpp::ttinfo::tt_isdst
```

Definition at line 382 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

### 6.791.2.4 tt\_ttisgmt

```
int Rcpp::ttinfo::tt_ttisgmt
```

Definition at line 385 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

### 6.791.2.5 tt\_ttisstd

```
int Rcpp::ttinfo::tt_ttisstd
```

Definition at line 384 of file date.cpp.

Referenced by Rcpp::typesequiv(), Rcpp::tzload(), and Rcpp::tzparse().

The documentation for this struct was generated from the following file:

- src/[date.cpp](#)

## 6.792 Rcpp::attributes::Type Class Reference

### Public Member Functions

- [Type](#) ()
- [Type](#) (const std::string &name, bool isConst, bool isReference)
- bool [empty](#) () const
- bool [operator==](#) (const [Type](#) &other) const
- bool [operator!=](#) (const [Type](#) &other) const
- const std::string & [name](#) () const
- std::string [full\\_name](#) () const
- bool [isVoid](#) () const
- bool [isConst](#) () const
- bool [isReference](#) () const

### Private Attributes

- std::string [name\\_](#)
- bool [isConst\\_](#)
- bool [isReference\\_](#)

### 6.792.1 Detailed Description

Definition at line 172 of file attributes.cpp.

### 6.792.2 Constructor & Destructor Documentation



### 6.792.2.1 Type() [1/2]

```
Rcpp::attributes::Type::Type ( ) [inline]
```

Definition at line 174 of file attributes.cpp.

### 6.792.2.2 Type() [2/2]

```
Rcpp::attributes::Type::Type (
    const std::string & name,
    bool isConst,
    bool isReference ) [inline]
```

Definition at line 175 of file attributes.cpp.

## 6.792.3 Member Function Documentation

### 6.792.3.1 empty()

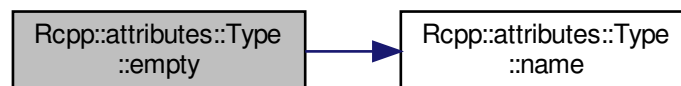
```
bool Rcpp::attributes::Type::empty ( ) const [inline]
```

Definition at line 179 of file attributes.cpp.

References name().

Referenced by Rcpp::attributes::Argument::empty(), and Rcpp::attributes::operator<<().

Here is the call graph for this function:



### 6.792.3.2 full\_name()

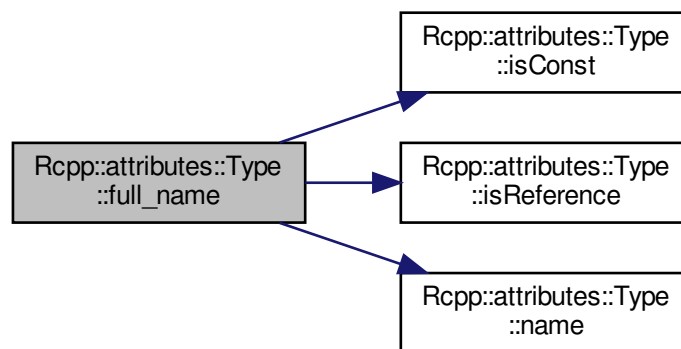
```
std::string Rcpp::attributes::Type::full_name ( ) const [inline]
```

Definition at line 192 of file attributes.cpp.

References isConst(), isReference(), and name().

Referenced by Rcpp::attributes::generateCpp().

Here is the call graph for this function:



### 6.792.3.3 isConst()

```
bool Rcpp::attributes::Type::isConst ( ) const [inline]
```

Definition at line 201 of file attributes.cpp.

References isConst\_.

Referenced by `full_name()`, and `Rcpp::attributes::operator<<()`.

### 6.792.3.4 isReference()

```
bool Rcpp::attributes::Type::isReference ( ) const [inline]
```

Definition at line 202 of file attributes.cpp.

References isReference\_.

Referenced by full\_name(), and Rcpp::attributes::operator<<().

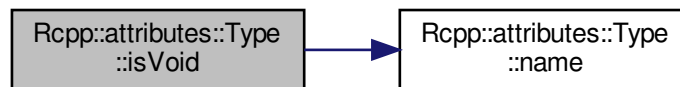
### 6.792.3.5 isVoid()

```
bool Rcpp::attributes::Type::isVoid ( ) const [inline]
```

Definition at line 200 of file attributes.cpp.

References name().

Here is the call graph for this function:



### 6.792.3.6 name()

```
const std::string& Rcpp::attributes::Type::name ( ) const [inline]
```

Definition at line 191 of file attributes.cpp.

References name\_.

Referenced by empty(), full\_name(), Rcpp::attributes::generateRArgList(), isVoid(), and Rcpp::attributes::operator<<().

### 6.792.3.7 operator"!=(())

```
bool Rcpp::attributes::Type::operator!= (
    const Type & other ) const [inline]
```

Definition at line 187 of file attributes.cpp.

### 6.792.3.8 operator==(())

```
bool Rcpp::attributes::Type::operator==(
    const Type & other ) const [inline]
```

Definition at line 181 of file attributes.cpp.

References isConst\_, isReference\_, and name\_.

## 6.792.4 Member Data Documentation

### 6.792.4.1 isConst\_

```
bool Rcpp::attributes::Type::isConst_ [private]
```

Definition at line 206 of file attributes.cpp.

Referenced by isConst(), and operator==(()).

### 6.792.4.2 isReference\_

```
bool Rcpp::attributes::Type::isReference_ [private]
```

Definition at line 207 of file attributes.cpp.

Referenced by isReference(), and operator==(()).

### 6.792.4.3 name\_

```
std::string Rcpp::attributes::Type::name_ [private]
```

Definition at line 205 of file attributes.cpp.

Referenced by name(), and operator==().

The documentation for this class was generated from the following file:

- [src/attributes.cpp](#)

## 6.793 Rcpp::tzhead Struct Reference

### Public Attributes

- char [tzh\\_magic](#) [4]
- char [tzh\\_version](#) [1]
- char [tzh\\_reserved](#) [15]
- char [tzh\\_ttisgmtcnt](#) [4]
- char [tzh\\_ttisstdcnt](#) [4]
- char [tzh\\_leapcnt](#) [4]
- char [tzh\\_timecnt](#) [4]
- char [tzh\\_typecnt](#) [4]
- char [tzh\\_charcnt](#) [4]

### 6.793.1 Detailed Description

Definition at line 205 of file date.cpp.

### 6.793.2 Member Data Documentation

#### 6.793.2.1 tzh\_charcnt

```
char Rcpp::tzhead::tzh_charcnt [4]
```

Definition at line 214 of file date.cpp.

### 6.793.2.2 tzh\_leapcnt

```
char Rcpp::tzhead::tzh_leapcnt[4]
```

Definition at line 211 of file date.cpp.

### 6.793.2.3 tzh\_magic

```
char Rcpp::tzhead::tzh_magic[4]
```

Definition at line 206 of file date.cpp.

### 6.793.2.4 tzh\_reserved

```
char Rcpp::tzhead::tzh_reserved[15]
```

Definition at line 208 of file date.cpp.

### 6.793.2.5 tzh\_timecnt

```
char Rcpp::tzhead::tzh_timecnt[4]
```

Definition at line 212 of file date.cpp.

### 6.793.2.6 tzh\_ttisgmtcnt

```
char Rcpp::tzhead::tzh_ttisgmtcnt[4]
```

Definition at line 209 of file date.cpp.

### 6.793.2.7 tzh\_ttisstdcnt

```
char Rcpp::tzhead::tzh_ttisstdcnt[4]
```

Definition at line 210 of file date.cpp.

### 6.793.2.8 tzh\_typecnt

```
char Rcpp::tzhead::tzh_typecnt[4]
```

Definition at line 213 of file date.cpp.

### 6.793.2.9 tzh\_version

```
char Rcpp::tzhead::tzh_version[1]
```

Definition at line 207 of file date.cpp.

The documentation for this struct was generated from the following file:

- [src/date.cpp](#)

## 6.794 Rcpp::traits::un\_pointer< T > Struct Template Reference

```
#include <un_pointer.h>
```

### Public Types

- typedef T [type](#)

### 6.794.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::un_pointer< T >
```

Definition at line 29 of file un\_pointer.h.

### 6.794.2 Member Typedef Documentation

### 6.794.2.1 type

```
template<typename T >
typedef T Rcpp::traits::un_pointer< T >::type
```

Definition at line 29 of file un\_pointer.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/un\_pointer.h

## 6.795 Rcpp::traits::un\_pointer< object< T > > Struct Template Reference

```
#include <un_pointer.h>
```

### Public Types

- typedef T [type](#)

### 6.795.1 Detailed Description

```
template<typename T>
struct Rcpp::traits::un_pointer< object< T > >
```

Definition at line 31 of file un\_pointer.h.

### 6.795.2 Member Typedef Documentation

#### 6.795.2.1 type

```
template<typename T >
typedef T Rcpp::traits::un_pointer< object< T > >::type
```

Definition at line 31 of file un\_pointer.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/un\_pointer.h



## 6.796 Rcpp::traits::un\_pointer< T \* > Struct Template Reference

```
#include <un_pointer.h>
```

### Public Types

- typedef T [type](#)

### 6.796.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::un_pointer< T * >
```

Definition at line 30 of file un\_pointer.h.

### 6.796.2 Member Typedef Documentation

#### 6.796.2.1 type

```
template<typename T >  
typedef T Rcpp::traits::un\_pointer< T \* >::type
```

Definition at line 30 of file un\_pointer.h.

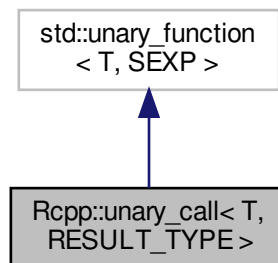
The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[un\\_pointer.h](#)

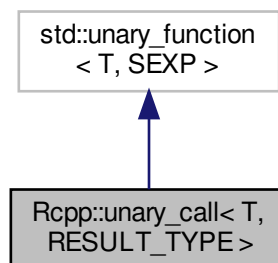
## 6.797 Rcpp::unary\_call< T, RESULT\_TYPE > Class Template Reference

```
#include <Language.h>
```

Inheritance diagram for Rcpp::unary\_call< T, RESULT\_TYPE >:



Collaboration diagram for Rcpp::unary\_call< T, RESULT\_TYPE >:



### Public Member Functions

- [unary\\_call](#) ([Language](#) call\_)
- [unary\\_call](#) ([Language](#) call\_, R\_xlen\_t index)
- [unary\\_call](#) ([Function](#) fun)
- RESULT\_TYPE [operator\(\)](#) (const T &object)

## Private Attributes

- [Language call](#)
- [Language::Proxy proxy](#)

### 6.797.1 Detailed Description

```
template<typename T, typename RESULT_TYPE = SEXP>  
class Rcpp::unary_call< T, RESULT_TYPE >
```

Definition at line 180 of file Language.h.

### 6.797.2 Constructor & Destructor Documentation

#### 6.797.2.1 unary\_call() [1/3]

```
template<typename T , typename RESULT_TYPE = SEXP>  
Rcpp::unary_call< T, RESULT_TYPE >::unary_call (  
    Language call_ ) [inline]
```

Definition at line 185 of file Language.h.

#### 6.797.2.2 unary\_call() [2/3]

```
template<typename T , typename RESULT_TYPE = SEXP>  
Rcpp::unary_call< T, RESULT_TYPE >::unary_call (  
    Language call_,  
    R_xlen_t index ) [inline]
```

Definition at line 186 of file Language.h.

#### 6.797.2.3 unary\_call() [3/3]

```
template<typename T , typename RESULT_TYPE = SEXP>  
Rcpp::unary_call< T, RESULT_TYPE >::unary_call (  
    Function fun ) [inline]
```

Definition at line 187 of file Language.h.

## 6.797.3 Member Function Documentation

### 6.797.3.1 operator>()

```
template<typename T , typename RESULT_TYPE = SEXP>
RESULT_TYPE Rcpp::unary_call< T, RESULT_TYPE >::operator() (
    const T & object ) [inline]
```

Definition at line 189 of file Language.h.

References Rcpp::unary\_call< T, RESULT\_TYPE >::call, and Rcpp::unary\_call< T, RESULT\_TYPE >::proxy.

## 6.797.4 Member Data Documentation

### 6.797.4.1 call

```
template<typename T , typename RESULT_TYPE = SEXP>
Language Rcpp::unary_call< T, RESULT_TYPE >::call [private]
```

Definition at line 195 of file Language.h.

Referenced by Rcpp::unary\_call< T, RESULT\_TYPE >::operator>(), and Rcpp::binary\_call< T1, T2, RESULT\_TYPE >::operator().

### 6.797.4.2 proxy

```
template<typename T , typename RESULT_TYPE = SEXP>
Language::Proxy Rcpp::unary_call< T, RESULT_TYPE >::proxy [private]
```

Definition at line 196 of file Language.h.

Referenced by Rcpp::unary\_call< T, RESULT\_TYPE >::operator().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/[Language.h](#)

## 6.798 Rcpp::sugar::unary\_minus< RTYPE, NA > Class Template Reference

```
#include <unary_minus.h>
```

### Public Types

- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [unary\\_minus\\_result\\_type](#)< RTYPE >::type [RESULT](#)

### Public Member Functions

- [RESULT](#) [apply](#) ([STORAGE](#) x) const

### 6.798.1 Detailed Description

```
template<int RTYPE, bool NA>  
class Rcpp::sugar::unary_minus< RTYPE, NA >
```

Definition at line 41 of file unary\_minus.h.

### 6.798.2 Member Typedef Documentation

#### 6.798.2.1 RESULT

```
template<int RTYPE, bool NA>  
typedef unary\_minus\_result\_type<RTYPE>::type Rcpp::sugar::unary\_minus< RTYPE, NA >::RESULT
```

Definition at line 44 of file unary\_minus.h.

#### 6.798.2.2 STORAGE

```
template<int RTYPE, bool NA>  
typedef traits::storage\_type<RTYPE>::type Rcpp::sugar::unary\_minus< RTYPE, NA >::STORAGE
```

Definition at line 43 of file unary\_minus.h.

## 6.798.3 Member Function Documentation

### 6.798.3.1 apply()

```
template<int RTYPE, bool NA>
RESULT Rcpp::sugar::unary_minus< RTYPE, NA >::apply (
    STORAGE x ) const [inline]
```

Definition at line 45 of file unary\_minus.h.

Referenced by Rcpp::sugar::UnaryMinus\_Vector< RTYPE, NA, T >::operator[]().

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/unary\_minus.h

## 6.799 Rcpp::sugar::unary\_minus< CPLXSCP, false > Class Reference

```
#include <unary_minus.h>
```

### Public Member Functions

- Rcomplex [apply](#) (Rcomplex x) const

### 6.799.1 Detailed Description

Definition at line 71 of file unary\_minus.h.

## 6.799.2 Member Function Documentation

### 6.799.2.1 apply()

```
Rcomplex Rcpp::sugar::unary_minus< CPLXSCP, false >::apply (
    Rcomplex x ) const [inline]
```

Definition at line 73 of file unary\_minus.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/unary\_minus.h

## 6.800 Rcpp::sugar::unary\_minus< CPLXSCP, NA > Class Template Reference

```
#include <unary_minus.h>
```

### Public Member Functions

- Rcomplex [apply](#) (Rcomplex x) const

### 6.800.1 Detailed Description

```
template<bool NA>  
class Rcpp::sugar::unary_minus< CPLXSCP, NA >
```

Definition at line 59 of file unary\_minus.h.

### 6.800.2 Member Function Documentation

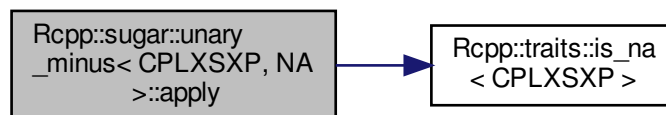
#### 6.800.2.1 apply()

```
template<bool NA>  
Rcomplex Rcpp::sugar::unary\_minus< CPLXSCP, NA >::apply (  
    Rcomplex x ) const [inline]
```

Definition at line 61 of file unary\_minus.h.

References [Rcpp::traits::is\\_na< CPLXSCP >\(\)](#).

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/[unary\\_minus.h](#)

## 6.801 Rcpp::sugar::unary\_minus< RTYPE, false > Class Template Reference

```
#include <unary_minus.h>
```

### Public Types

- typedef [Rcpp::traits::storage\\_type< RTYPE >::type](#) [STORAGE](#)
- typedef [unary\\_minus\\_result\\_type< RTYPE >::type](#) [RESULT](#)

### Public Member Functions

- [RESULT](#) [apply](#) ([STORAGE](#) x) const

#### 6.801.1 Detailed Description

```
template<int RTYPE>  
class Rcpp::sugar::unary_minus< RTYPE, false >
```

Definition at line 50 of file unary\_minus.h.

#### 6.801.2 Member Typedef Documentation

##### 6.801.2.1 RESULT

```
template<int RTYPE>  
typedef unary\_minus\_result\_type<RTYPE>::type Rcpp::sugar::unary\_minus< RTYPE, false >::RESULT
```

Definition at line 53 of file unary\_minus.h.

##### 6.801.2.2 STORAGE

```
template<int RTYPE>  
typedef Rcpp::traits::storage\_type<RTYPE>::type Rcpp::sugar::unary\_minus< RTYPE, false >::STORAGE
```

Definition at line 52 of file unary\_minus.h.



## 6.801.3 Member Function Documentation

### 6.801.3.1 apply()

```
template<int RTYPE>
RESULT Rcpp::sugar::unary_minus< RTYPE, false >::apply (
    STORAGE x ) const [inline]
```

Definition at line 54 of file unary\_minus.h.

The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/unary\_minus.h

## 6.802 Rcpp::sugar::unary\_minus\_result\_type< RTYPE > Struct Template Reference

```
#include <unary_minus.h>
```

### Public Types

- enum { value = RTYPE }
- typedef traits::storage\_type< RTYPE >::type type

### 6.802.1 Detailed Description

```
template<int RTYPE>
struct Rcpp::sugar::unary_minus_result_type< RTYPE >
```

Definition at line 29 of file unary\_minus.h.

### 6.802.2 Member Typedef Documentation

#### 6.802.2.1 type

```
template<int RTYPE>
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::unary_minus_result_type< RTYPE >::type
```

Definition at line 30 of file unary\_minus.h.

### 6.802.3 Member Enumeration Documentation

#### 6.802.3.1 anonymous enum

```
template<int RTYPE>
anonymous enum
```

##### Enumerator

value	
-------	--

Definition at line 31 of file unary\_minus.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/operators/unary\\_minus.h](#)

### 6.803 Rcpp::sugar::unary\_minus\_result\_type< LGLSXP > Struct Reference

```
#include <unary_minus.h>
```

#### Public Types

- enum { [value](#) = INTSXP }
- typedef [traits::storage\\_type](#)< INTSXP >::[type](#) [type](#)

#### 6.803.1 Detailed Description

Definition at line 34 of file unary\_minus.h.

#### 6.803.2 Member Typedef Documentation

##### 6.803.2.1 type

```
typedef traits::storage\_type<INTSXP>::type Rcpp::sugar::unary\_minus\_result\_type< LGLSXP >::type
```

Definition at line 35 of file unary\_minus.h.

### 6.803.3 Member Enumeration Documentation

#### 6.803.3.1 anonymous enum

anonymous enum

Enumerator

value	
-------	--

Definition at line 36 of file unary\_minus.h.

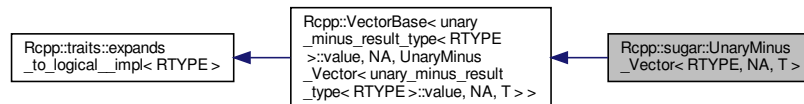
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/sugar/operators/unary\\_minus.h](inst/include/Rcpp/sugar/operators/unary_minus.h)

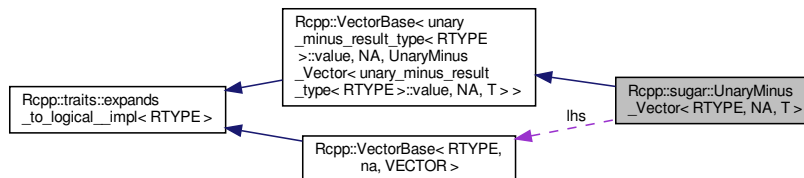
## 6.804 Rcpp::sugar::UnaryMinus\_Vector< RTYPE, NA, T > Class Template Reference

```
#include <unary_minus.h>
```

Inheritance diagram for Rcpp::sugar::UnaryMinus\_Vector< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::UnaryMinus\_Vector< RTYPE, NA, T >:



## Public Types

- typedef [Rcpp::VectorBase](#)< RTYPE, NA, T > [VEC\\_TYPE](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)
- typedef [unary\\_minus\\_result\\_type](#)< RTYPE >::type [RESULT](#)
- typedef [unary\\_minus](#)< RTYPE, NA > [OPERATOR](#)

## Public Member Functions

- [UnaryMinus\\_Vector](#) (const [VEC\\_TYPE](#) &lhs\_)
- [RESULT operator\[\]](#) (R\_xlen\_t i) const
- [R\\_xlen\\_t size](#) () const

## Private Attributes

- const [VEC\\_TYPE](#) & lhs
- [OPERATOR](#) op

### 6.804.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >
```

Definition at line 83 of file unary\_minus.h.

### 6.804.2 Member Typedef Documentation

#### 6.804.2.1 OPERATOR

```
template<int RTYPE, bool NA, typename T >
typedef unary\_minus<RTYPE,NA> Rcpp::sugar::UnaryMinus\_Vector< RTYPE, NA, T >::OPERATOR
```

Definition at line 92 of file unary\_minus.h.

#### 6.804.2.2 RESULT

```
template<int RTYPE, bool NA, typename T >
typedef unary\_minus\_result\_type<RTYPE>::type Rcpp::sugar::UnaryMinus\_Vector< RTYPE, NA, T >↔
::RESULT
```

Definition at line 91 of file unary\_minus.h.

### 6.804.2.3 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef traits::storage_type<RTYPE>::type Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::STORAGE
```

Definition at line 90 of file unary\_minus.h.

### 6.804.2.4 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 89 of file unary\_minus.h.

## 6.804.3 Constructor & Destructor Documentation

### 6.804.3.1 UnaryMinus\_Vector()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::UnaryMinus_Vector (
    const VEC_TYPE & lhs_ ) [inline]
```

Definition at line 94 of file unary\_minus.h.

### 6.804.4 Member Function Documentation

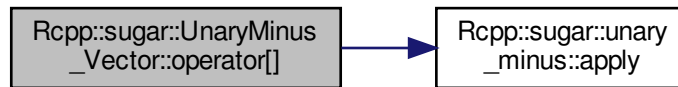
### 6.804.4.1 operator[]()

```
template<int RTYPE, bool NA, typename T >
RESULT Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 97 of file unary\_minus.h.

References Rcpp::sugar::unary\_minus< RTYPE, NA >::apply(), Rcpp::sugar::UnaryMinus\_Vector< RTYPE, NA, T >::lhs, and Rcpp::sugar::UnaryMinus\_Vector< RTYPE, NA, T >::op.

Here is the call graph for this function:



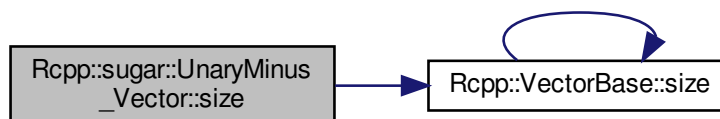
### 6.804.4.2 size()

```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 101 of file unary\_minus.h.

References Rcpp::sugar::UnaryMinus\_Vector< RTYPE, NA, T >::lhs, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



## 6.804.5 Member Data Documentation

### 6.804.5.1 lhs

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::lhs [private]
```

Definition at line 104 of file unary\_minus.h.

Referenced by Rcpp::sugar::UnaryMinus\_Vector< RTYPE, NA, T >::operator[](), and Rcpp::sugar::UnaryMinus\_←Vector< RTYPE, NA, T >::size().

### 6.804.5.2 op

```
template<int RTYPE, bool NA, typename T >
OPERATOR Rcpp::sugar::UnaryMinus_Vector< RTYPE, NA, T >::op [private]
```

Definition at line 105 of file unary\_minus.h.

Referenced by Rcpp::sugar::UnaryMinus\_Vector< RTYPE, NA, T >::operator[]().

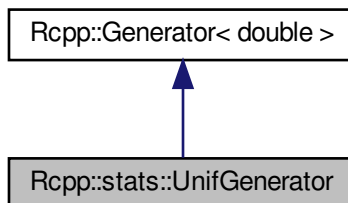
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/operators/unary\_minus.h

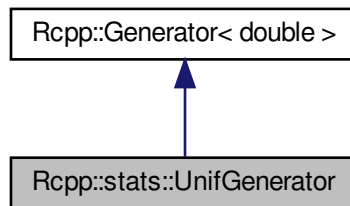
## 6.805 Rcpp::stats::UnifGenerator Class Reference

```
#include <runif.h>
```

Inheritance diagram for Rcpp::stats::UnifGenerator:



Collaboration diagram for Rcpp::stats::UnifGenerator:



## Public Member Functions

- [UnifGenerator](#) (double min\_=0.0, double max\_=1.0)
- double [operator\(\)](#) () const

## Private Attributes

- double [min](#)
- double [diff](#)

## Additional Inherited Members

### 6.805.1 Detailed Description

Definition at line 28 of file runif.h.

### 6.805.2 Constructor & Destructor Documentation

#### 6.805.2.1 UnifGenerator()

```
Rcpp::stats::UnifGenerator::UnifGenerator (
    double min_ = 0.0,
    double max_ = 1.0 ) [inline]
```

Definition at line 31 of file runif.h.



## 6.805.3 Member Function Documentation

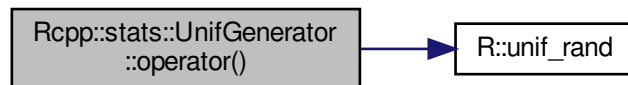
### 6.805.3.1 operator()

```
double Rcpp::stats::UnifGenerator::operator() ( ) const [inline]
```

Definition at line 34 of file runif.h.

References `diff`, `min`, and `R::unif_rand()`.

Here is the call graph for this function:



## 6.805.4 Member Data Documentation

### 6.805.4.1 diff

```
double Rcpp::stats::UnifGenerator::diff [private]
```

Definition at line 42 of file runif.h.

Referenced by `operator()`.

### 6.805.4.2 min

```
double Rcpp::stats::UnifGenerator::min [private]
```

Definition at line 41 of file runif.h.

Referenced by `operator()`.

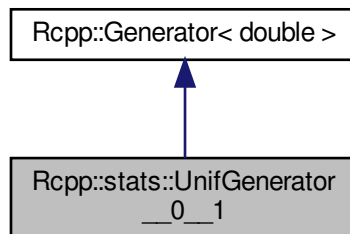
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/runif.h](#)

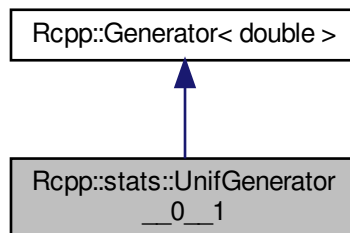
## 6.806 Rcpp::stats::UnifGenerator\_\_0\_\_1 Class Reference

```
#include <runif.h>
```

Inheritance diagram for Rcpp::stats::UnifGenerator\_\_0\_\_1:



Collaboration diagram for Rcpp::stats::UnifGenerator\_\_0\_\_1:



### Public Member Functions

- [UnifGenerator\\_\\_0\\_\\_1 \(\)](#)
- double [operator\(\) \(\)](#) const

### Additional Inherited Members

#### 6.806.1 Detailed Description

Definition at line 45 of file runif.h.

## 6.806.2 Constructor & Destructor Documentation

### 6.806.2.1 UnifGenerator\_\_0\_\_1()

```
Rcpp::stats::UnifGenerator__0__1::UnifGenerator__0__1 ( ) [inline]
```

Definition at line 48 of file runif.h.

## 6.806.3 Member Function Documentation

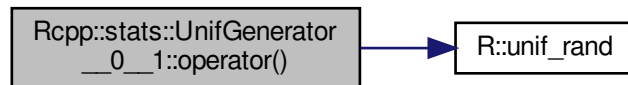
### 6.806.3.1 operator>()

```
double Rcpp::stats::UnifGenerator__0__1::operator() ( ) const [inline]
```

Definition at line 50 of file runif.h.

References R::unif\_rand().

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/runif.h](#)

## 6.807 Rcpp::sugar::Union< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T > Class Template Reference

```
#include <setdiff.h>
```

## Public Types

- typedef [Rcpp::traits::storage\\_type](#)< RTYPE >::type [STORAGE](#)

## Public Member Functions

- [Union](#) (const LHS\_T &lhs, const RHS\_T &rhs)
- [Vector](#)< RTYPE > [get](#) () const

## Private Types

- typedef [RCPP\\_UNORDERED\\_SET](#)< [STORAGE](#) > [SET](#)
- typedef [SET](#)::const\_iterator [ITERATOR](#)

## Private Attributes

- [SET](#) result

### 6.807.1 Detailed Description

```
template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T>
class Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >
```

Definition at line 133 of file setdiff.h.

### 6.807.2 Member Typedef Documentation

#### 6.807.2.1 ITERATOR

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef SET::const_iterator Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::ITERATOR
[private]
```

Definition at line 152 of file setdiff.h.

### 6.807.2.2 SET

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef RCPP_UNORDERED_SET<STORAGE> Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >↔
::SET [private]
```

Definition at line 151 of file setdiff.h.

### 6.807.2.3 STORAGE

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA,
RHS_T >::STORAGE
```

Definition at line 135 of file setdiff.h.

## 6.807.3 Constructor & Destructor Documentation

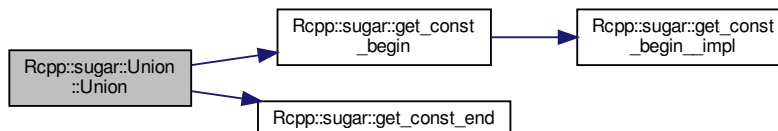
### 6.807.3.1 Union()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::Union (
    const LHS_T & lhs,
    const RHS_T & rhs ) [inline]
```

Definition at line 137 of file setdiff.h.

References [Rcpp::sugar::get\\_const\\_begin\(\)](#), and [Rcpp::sugar::get\\_const\\_end\(\)](#).

Here is the call graph for this function:



## 6.807.4 Member Function Documentation

### 6.807.4.1 get()

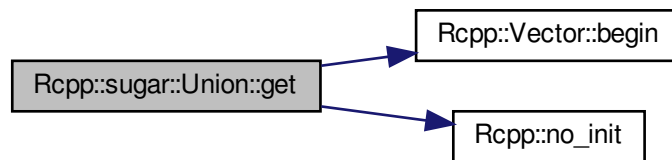
```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Vector<RTYPE> Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::get ( ) const [inline]
```

Definition at line 143 of file setdiff.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::no\_init().

Referenced by Rcpp::union\_().

Here is the call graph for this function:



## 6.807.5 Member Data Documentation

### 6.807.5.1 result

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
SET Rcpp::sugar::Union< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >::result [private]
```

Definition at line 153 of file setdiff.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/setdiff.h](#)

## 6.808 Rcpp::internal::UnwindData Struct Reference

```
#include <unwindProtect.h>
```

### Public Attributes

- `std::jmp_buf` [jmpbuf](#)

### 6.808.1 Detailed Description

Definition at line 34 of file `unwindProtect.h`.

### 6.808.2 Member Data Documentation

#### 6.808.2.1 jmpbuf

```
std::jmp_buf Rcpp::internal::UnwindData::jmpbuf
```

Definition at line 35 of file `unwindProtect.h`.

Referenced by `Rcpp::internal::maybeJump()`, and `Rcpp::unwindProtect()`.

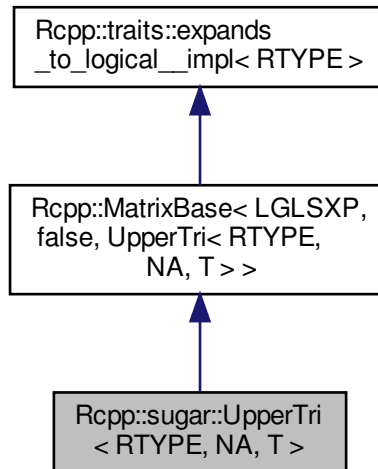
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/unwindProtect.h`

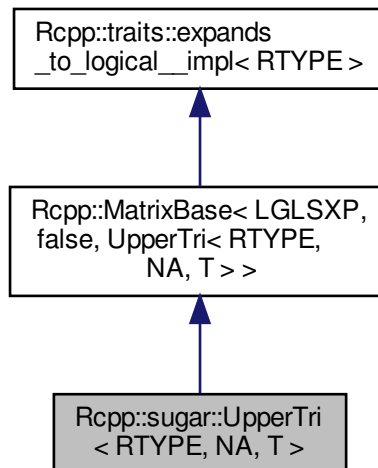
## 6.809 Rcpp::sugar::UpperTri< RTYPE, NA, T > Class Template Reference

```
#include <upper_tri.h>
```

Inheritance diagram for `Rcpp::sugar::UpperTri< RTYPE, NA, T >`:



Collaboration diagram for `Rcpp::sugar::UpperTri< RTYPE, NA, T >`:



## Public Types

- typedef `Rcpp::MatrixBase< RTYPE, NA, T >` `MatBase`



## Public Member Functions

- [UpperTri](#) (const T &lhs, bool [diag](#))
- int [operator\(\)](#) (int i, int j) const
- R\_xlen\_t [size](#) () const
- int [nrow](#) () const
- int [ncol](#) () const

## Private Types

- typedef bool(UpperTri::\* [Method](#)) (int, int) const

## Private Member Functions

- bool [get\\_diag\\_true](#) (int i, int j) const
- bool [get\\_diag\\_false](#) (int i, int j) const
- bool [get](#) (int i, int j) const

## Private Attributes

- int [nr](#)
- int [nc](#)
- [Method](#) [getter](#)

## Additional Inherited Members

### 6.809.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::UpperTri< RTYPE, NA, T >
```

Definition at line 30 of file upper\_tri.h.

### 6.809.2 Member Typedef Documentation

#### 6.809.2.1 MatBase

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::MatrixBase<RTYPE, NA, T> Rcpp::sugar::UpperTri< RTYPE, NA, T >::MatBase
```

Definition at line 32 of file upper\_tri.h.

### 6.809.2.2 Method

```
template<int RTYPE, bool NA, typename T >
typedef bool(UpperTri::* Rcpp::sugar::UpperTri< RTYPE, NA, T >::Method) (int, int) const [private]
```

Definition at line 47 of file upper\_tri.h.

## 6.809.3 Constructor & Destructor Documentation

### 6.809.3.1 UpperTri()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::UpperTri< RTYPE, NA, T >::UpperTri (
    const T & lhs,
    bool diag ) [inline]
```

Definition at line 34 of file upper\_tri.h.

## 6.809.4 Member Function Documentation

### 6.809.4.1 get()

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::UpperTri< RTYPE, NA, T >::get (
    int i,
    int j ) const [inline], [private]
```

Definition at line 56 of file upper\_tri.h.

Referenced by Rcpp::sugar::UpperTri< RTYPE, NA, T >::operator()().

### 6.809.4.2 get\_diag\_false()

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::UpperTri< RTYPE, NA, T >::get_diag_false (
    int i,
    int j ) const [inline], [private]
```

Definition at line 54 of file upper\_tri.h.

### 6.809.4.3 get\_diag\_true()

```
template<int RTYPE, bool NA, typename T >
bool Rcpp::sugar::UpperTri< RTYPE, NA, T >::get_diag_true (
    int i,
    int j ) const [inline], [private]
```

Definition at line 52 of file upper\_tri.h.

### 6.809.4.4 ncol()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::UpperTri< RTYPE, NA, T >::ncol ( ) const [inline]
```

Definition at line 44 of file upper\_tri.h.

References Rcpp::sugar::UpperTri< RTYPE, NA, T >::nc.

### 6.809.4.5 nrow()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::UpperTri< RTYPE, NA, T >::nrow ( ) const [inline]
```

Definition at line 43 of file upper\_tri.h.

References Rcpp::sugar::UpperTri< RTYPE, NA, T >::nr.

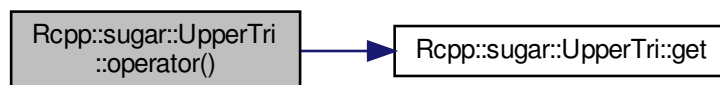
### 6.809.4.6 operator()()

```
template<int RTYPE, bool NA, typename T >
int Rcpp::sugar::UpperTri< RTYPE, NA, T >::operator() (
    int i,
    int j ) const [inline]
```

Definition at line 40 of file upper\_tri.h.

References Rcpp::sugar::UpperTri< RTYPE, NA, T >::get().

Here is the call graph for this function:



### 6.809.4.7 size()

```
template<int RTYPE, bool NA, typename T >  
R_xlen_t Rcpp::sugar::UpperTri< RTYPE, NA, T >::size ( ) const [inline]
```

Definition at line 42 of file upper\_tri.h.

References `Rcpp::sugar::UpperTri< RTYPE, NA, T >::nc`, and `Rcpp::sugar::UpperTri< RTYPE, NA, T >::nr`.

## 6.809.5 Member Data Documentation

### 6.809.5.1 getter

```
template<int RTYPE, bool NA, typename T >  
Method Rcpp::sugar::UpperTri< RTYPE, NA, T >::getter [private]
```

Definition at line 50 of file upper\_tri.h.

### 6.809.5.2 nc

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::sugar::UpperTri< RTYPE, NA, T >::nc [private]
```

Definition at line 49 of file upper\_tri.h.

Referenced by `Rcpp::sugar::UpperTri< RTYPE, NA, T >::ncol()`, and `Rcpp::sugar::UpperTri< RTYPE, NA, T >::size()`.

### 6.809.5.3 nr

```
template<int RTYPE, bool NA, typename T >  
int Rcpp::sugar::UpperTri< RTYPE, NA, T >::nr [private]
```

Definition at line 49 of file upper\_tri.h.

Referenced by `Rcpp::sugar::UpperTri< RTYPE, NA, T >::nrow()`, and `Rcpp::sugar::UpperTri< RTYPE, NA, T >::size()`.

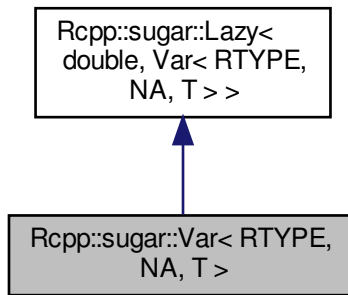
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/matrix/upper\\_tri.h](#)

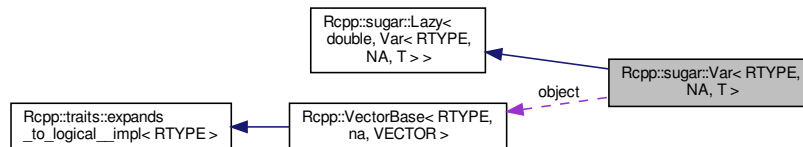
## 6.810 Rcpp::sugar::Var< RTYPE, NA, T > Class Template Reference

```
#include <var.h>
```

Inheritance diagram for Rcpp::sugar::Var< RTYPE, NA, T >:



Collaboration diagram for Rcpp::sugar::Var< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`

### Public Member Functions

- `Var` (const `VEC_TYPE` &object\_)
- `double get ()` const

### Private Attributes

- const `VEC_TYPE` & `object`

### 6.810.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>  
class Rcpp::sugar::Var< RTYPE, NA, T >
```

Definition at line 30 of file var.h.

### 6.810.2 Member Typedef Documentation

#### 6.810.2.1 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >  
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::Var< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 32 of file var.h.

### 6.810.3 Constructor & Destructor Documentation

#### 6.810.3.1 Var()

```
template<int RTYPE, bool NA, typename T >  
Rcpp::sugar::Var< RTYPE, NA, T >::Var (  
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 34 of file var.h.

### 6.810.4 Member Function Documentation

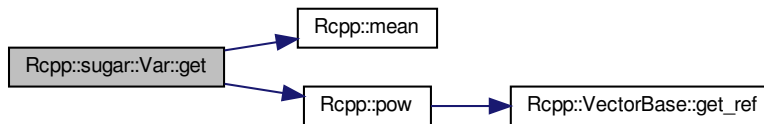
### 6.810.4.1 get()

```
template<int RTYPE, bool NA, typename T >  
double Rcpp::sugar::Var< RTYPE, NA, T >::get ( ) const [inline]
```

Definition at line 36 of file var.h.

References Rcpp::mean(), and Rcpp::pow().

Here is the call graph for this function:



## 6.810.5 Member Data Documentation

### 6.810.5.1 object

```
template<int RTYPE, bool NA, typename T >  
const VEC_TYPE& Rcpp::sugar::Var< RTYPE, NA, T >::object [private]
```

Definition at line 46 of file var.h.

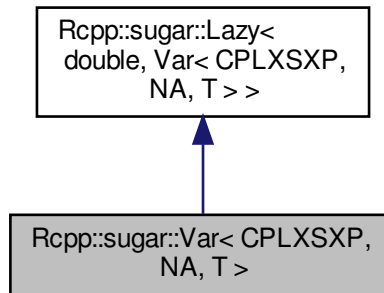
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/var.h

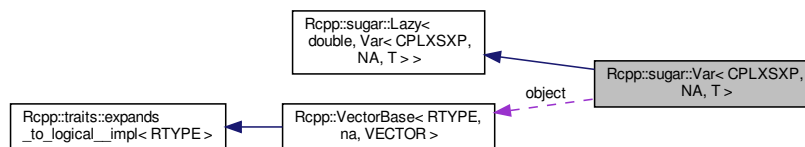
## 6.811 Rcpp::sugar::Var< CPLXSP, NA, T > Class Template Reference

```
#include <var.h>
```

Inheritance diagram for Rcpp::sugar::Var< CPLXSP, NA, T >:



Collaboration diagram for Rcpp::sugar::Var< CPLXSP, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< CPLXSP, NA, T >` `VEC_TYPE`

### Public Member Functions

- `Var` (const `VEC_TYPE` &object\_)
- `double get ()` const

### Private Attributes

- const `VEC_TYPE` & `object`



## 6.811.1 Detailed Description

```
template<bool NA, typename T>
class Rcpp::sugar::Var< CPLXSXP, NA, T >
```

Definition at line 50 of file var.h.

## 6.811.2 Member Typedef Documentation

### 6.811.2.1 VEC\_TYPE

```
template<bool NA, typename T >
typedef Rcpp::VectorBase<CPLXSXP,NA,T> Rcpp::sugar::Var< CPLXSXP, NA, T >::VEC_TYPE
```

Definition at line 52 of file var.h.

## 6.811.3 Constructor & Destructor Documentation

### 6.811.3.1 Var()

```
template<bool NA, typename T >
Rcpp::sugar::Var< CPLXSXP, NA, T >::Var (
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 54 of file var.h.

## 6.811.4 Member Function Documentation

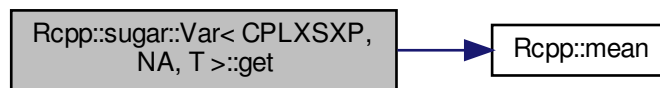
### 6.811.4.1 get()

```
template<bool NA, typename T >  
double Rcpp::sugar::Var< CPLXSXP, NA, T >::get ( ) const [inline]
```

Definition at line 56 of file var.h.

References Rcpp::mean().

Here is the call graph for this function:



## 6.811.5 Member Data Documentation

### 6.811.5.1 object

```
template<bool NA, typename T >  
const VEC_TYPE& Rcpp::sugar::Var< CPLXSXP, NA, T >::object [private]
```

Definition at line 68 of file var.h.

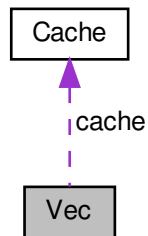
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/functions/var.h](#)

## 6.812 Vec Class Reference

```
#include <convolve10_cpp.h>
```

Collaboration diagram for Vec:



### Public Types

- typedef double & [proxy](#)
- typedef double & [proxy](#)

### Public Member Functions

- [Vec](#) (double \*data\_)
- [proxy operator\[\]](#) (int i)
- [proxy operator\[\]](#) (int i) const
- [Vec](#) (double \*data\_)
- double & [operator\[\]](#) (int i)
- [Vec](#) (double \*data\_)
- [proxy operator\[\]](#) (int i)
- [proxy operator\[\]](#) (int i) const

### Private Attributes

- [Cache](#) [cache](#)
- double \* [data](#)

### 6.812.1 Detailed Description

Definition at line 16 of file convolve10\_cpp.h.

## 6.812.2 Member Typedef Documentation

### 6.812.2.1 proxy [1/2]

```
typedef double& Vec::proxy
```

Definition at line 18 of file convolve10\_cpp.h.

### 6.812.2.2 proxy [2/2]

```
typedef double& Vec::proxy
```

Definition at line 31 of file convolve9\_cpp.cpp.

## 6.812.3 Constructor & Destructor Documentation

### 6.812.3.1 Vec() [1/3]

```
Vec::Vec (  
    double * data_ ) [inline]
```

Definition at line 20 of file convolve10\_cpp.h.

### 6.812.3.2 Vec() [2/3]

```
Vec::Vec (  
    double * data_ ) [inline]
```

Definition at line 16 of file convolve8\_cpp.cpp.

### 6.812.3.3 Vec() [3/3]

```
Vec::Vec (
    double * data_ ) [inline]
```

Definition at line 33 of file convolve9\_cpp.cpp.

## 6.812.4 Member Function Documentation

### 6.812.4.1 operator[]() [1/5]

```
proxy Vec::operator[] (
    int i ) [inline]
```

Definition at line 21 of file convolve10\_cpp.h.

References `cache`, and `Cache::ref()`.

Here is the call graph for this function:



### 6.812.4.2 operator[]() [2/5]

```
double& Vec::operator[] (
    int i ) [inline]
```

Definition at line 17 of file convolve8\_cpp.cpp.

References `data`.

### 6.812.4.3 operator[]() [3/5]

```
proxy Vec::operator[] (
    int i ) [inline]
```

Definition at line 34 of file convolve9\_cpp.cpp.

References cache, and Cache::ref().

Here is the call graph for this function:



### 6.812.4.4 operator[]() [4/5]

```
proxy Vec::operator[] (
    int i ) const [inline]
```

Definition at line 22 of file convolve10\_cpp.h.

References cache, and Cache::ref().

Here is the call graph for this function:



### 6.812.4.5 operator[]() [5/5]

```
proxy Vec::operator[] (
    int i ) const [inline]
```

Definition at line 35 of file convolve9\_cpp.cpp.

References `cache`, and `Cache::ref()`.

Here is the call graph for this function:



## 6.812.5 Member Data Documentation

### 6.812.5.1 cache

```
Cache Vec::cache [private]
```

Definition at line 25 of file convolve10\_cpp.h.

Referenced by `operator[]()`.

### 6.812.5.2 data

```
double* Vec::data [private]
```

Definition at line 20 of file convolve8\_cpp.cpp.

Referenced by `operator[]()`.

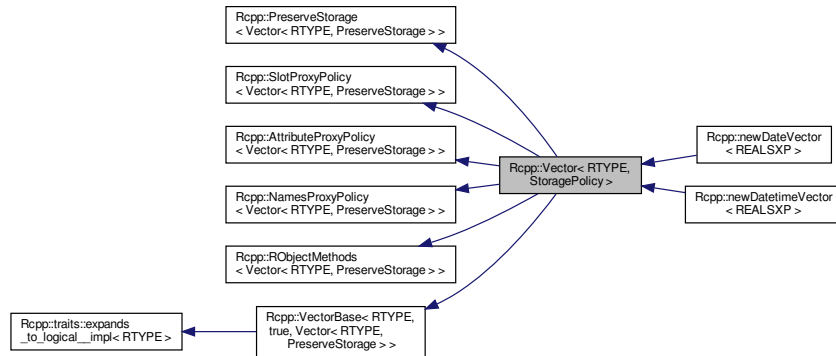
The documentation for this class was generated from the following files:

- [inst/examples/ConvolveBenchmarks/convolve10\\_cpp.h](#)
- [inst/examples/ConvolveBenchmarks/convolve9\\_cpp.cpp](#)
- [inst/examples/ConvolveBenchmarks/convolve8\\_cpp.cpp](#)

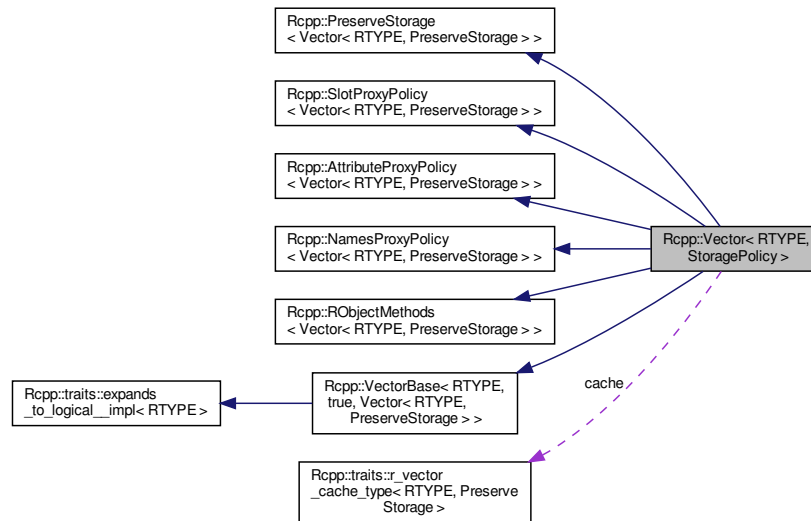
## 6.813 Rcpp::Vector< RTYPE, StoragePolicy > Class Template Reference

```
#include <Vector.h>
```

Inheritance diagram for Rcpp::Vector< RTYPE, StoragePolicy >:



Collaboration diagram for Rcpp::Vector< RTYPE, StoragePolicy >:



### Public Types

- typedef StoragePolicy< Vector > Storage
- typedef traits::r\_vector\_proxy< RTYPE, StoragePolicy >::type Proxy
- typedef traits::r\_vector\_const\_proxy< RTYPE, StoragePolicy >::type const\_Proxy



- typedef [traits::r\\_vector\\_name\\_proxy](#)< RTYPE, StoragePolicy >::type [NameProxy](#)
- typedef [traits::r\\_vector\\_proxy](#)< RTYPE, StoragePolicy >::type [value\\_type](#)
- typedef [traits::r\\_vector\\_iterator](#)< RTYPE, StoragePolicy >::type [iterator](#)
- typedef [traits::r\\_vector\\_const\\_iterator](#)< RTYPE, StoragePolicy >::type [const\\_iterator](#)
- typedef [traits::init\\_type](#)< RTYPE >::type [init\\_type](#)
- typedef [traits::r\\_vector\\_element\\_converter](#)< RTYPE >::type [converter\\_type](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [stored\\_type](#)
- typedef [internal::RangeIndexer](#)< RTYPE, true, [Vector](#) > [Indexer](#)

## Public Member Functions

- [Vector](#) ()
- [Vector](#) (const [Vector](#) &other)
- [Vector](#) & operator= (const [Vector](#) &rhs)
- [Vector](#) (SEXP x)
- template<typename Proxy >  
[Vector](#) (const [GenericProxy](#)< Proxy > &proxy)
- [Vector](#) (const [no\\_init\\_vector](#) &obj)
- template<typename T >  
[Vector](#) (const T &size, const [stored\\_type](#) &u, typename [Rcpp::traits::enable\\_if](#)< [traits::is\\_arithmetic](#)< T >::value, void >::type \*=0)
- [Vector](#) (const int &size, const [stored\\_type](#) &u)
- [Vector](#) (const std::string &st)
- [Vector](#) (const char \*st)
- template<typename T >  
[Vector](#) (const T &size, [stored\\_type](#)(\*gen)(void), typename [Rcpp::traits::enable\\_if](#)< [traits::is\\_arithmetic](#)< T >::value, void >::type \*=0)
- template<typename T >  
[Vector](#) (T size, typename [Rcpp::traits::enable\\_if](#)< [traits::is\\_arithmetic](#)< T >::value, void >::type \*=0)
- [Vector](#) (const int &size)
- [Vector](#) (const [Dimension](#) &dims)
- template<typename T >  
[Vector](#) (T value, typename [Rcpp::traits::enable\\_if](#)< [traits::is\\_bool](#)< T >::value &&RTYPE==LGLSXP, void >::type \*=0)
- template<typename U >  
[Vector](#) (const [Dimension](#) &dims, const U &u)
- template<bool NA, typename VEC >  
[Vector](#) (const [VectorBase](#)< RTYPE, NA, VEC > &other)
- template<typename T, typename U >  
[Vector](#) (const T &size, const U &u, typename [Rcpp::traits::enable\\_if](#)< [traits::is\\_arithmetic](#)< T >::value, void >::type \*=0)
- template<bool NA, typename T >  
[Vector](#) (const [sugar::SingleLogicalResult](#)< NA, T > &obj)
- template<typename T, typename U1 >  
[Vector](#) (const T &size, [stored\\_type](#)(\*gen)(U1), const U1 &u1, typename [Rcpp::traits::enable\\_if](#)< [traits::is\\_arithmetic](#)< T >::value, void >::type \*=0)
- template<typename T, typename U1, typename U2 >  
[Vector](#) (const T &size, [stored\\_type](#)(\*gen)(U1, U2), const U1 &u1, const U2 &u2, typename [Rcpp::traits::enable\\_if](#)< [traits::is\\_arithmetic](#)< T >::value, void >::type \*=0)

- `template<typename T, typename U1, typename U2, typename U3 >`  
[Vector](#) (const T &sz, [stored\\_type](#)(\*gen)(U1, U2, U3), const U1 &u1, const U2 &u2, const U3 &u3, typename [Rcpp::traits::enable\\_if](#)< [traits::is\\_arithmetic](#)< T >::value, void >::type \*=0)
- `template<typename InputIterator >`  
[Vector](#) (InputIterator first, InputIterator last)
- `template<typename InputIterator, typename T >`  
[Vector](#) (InputIterator first, InputIterator last, T n, typename [Rcpp::traits::enable\\_if](#)< [traits::is\\_arithmetic](#)< T >::value, void >::type \*=0)
- `template<typename InputIterator, typename Func >`  
[Vector](#) (InputIterator first, InputIterator last, Func func)
- `template<typename InputIterator, typename Func, typename T >`  
[Vector](#) (InputIterator first, InputIterator last, Func func, T n, typename [Rcpp::traits::enable\\_if](#)< [traits::is\\_arithmetic](#)< T >::value, void >::type \*=0)
- `template<typename T >`  
[Vector](#) & [operator=](#) (const T &x)
- [R\\_xlen\\_t length](#) () const
- [R\\_xlen\\_t size](#) () const
- [R\\_xlen\\_t offset](#) (const int &i, const int &j) const
- [R\\_xlen\\_t offset](#) (const [R\\_xlen\\_t](#) &i) const
- [R\\_xlen\\_t offset](#) (const std::string &name) const
- `template<typename U >`  
void [fill](#) (const U &u)
- [iterator begin](#) ()
- [iterator end](#) ()
- [const\\_iterator begin](#) () const
- [const\\_iterator end](#) () const
- [const\\_iterator cbegin](#) () const
- [const\\_iterator cend](#) () const
- [Proxy operator\[\]](#) ([R\\_xlen\\_t](#) i)
- [const\\_Proxy operator\[\]](#) ([R\\_xlen\\_t](#) i) const
- [Proxy operator\(\)](#) (const [size\\_t](#) &i)
- [const\\_Proxy operator\(\)](#) (const [size\\_t](#) &i) const
- [Proxy at](#) (const [size\\_t](#) &i)
- [const\\_Proxy at](#) (const [size\\_t](#) &i) const
- [Proxy operator\(\)](#) (const [size\\_t](#) &i, const [size\\_t](#) &j)
- [const\\_Proxy operator\(\)](#) (const [size\\_t](#) &i, const [size\\_t](#) &j) const
- [NameProxy operator\[\]](#) (const std::string &name)
- [NameProxy operator\(\)](#) (const std::string &name)
- [NameProxy operator\[\]](#) (const std::string &name) const
- [NameProxy operator\(\)](#) (const std::string &name) const
- [operator RObject](#) () const
- `template<int RHS_RTYPE, bool RHS_NA, typename RHS_T >`  
[SubsetProxy](#)< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T > [operator\[\]](#) (const [VectorBase](#)< RHS\_RTYPE, RHS\_NA, RHS\_T > &rhs)
- `template<int RHS_RTYPE, bool RHS_NA, typename RHS_T >`  
const [SubsetProxy](#)< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T > [operator\[\]](#) (const [VectorBase](#)< RHS\_RTYPE, RHS\_NA, RHS\_T > &rhs) const
- [Vector](#) & [sort](#) (bool decreasing=false)
- `template<typename InputIterator >`  
void [assign](#) (InputIterator first, InputIterator last)
- `template<typename T >`  
void [push\\_back](#) (const T &object)

- `template<typename T >`  
void `push_back` (const T &`object`, const std::string &`name`)
- `template<typename T >`  
void `push_front` (const T &`object`)
- `template<typename T >`  
void `push_front` (const T &`object`, const std::string &`name`)
- `template<typename T >`  
`iterator insert` (`iterator` `position`, const T &`object`)
- `template<typename T >`  
`iterator insert` (int `position`, const T &`object`)
- `iterator erase` (int `position`)
- `iterator erase` (`iterator` `position`)
- `iterator erase` (int `first`, int `last`)
- `iterator erase` (`iterator` `first`, `iterator` `last`)
- void `update` (SEXP)
- `Indexer operator[]` (const `Range` &`range`)
- `template<typename EXPR_VEC >`  
`Vector & operator+=` (const `VectorBase`< RTYPE, true, EXPR\_VEC > &`rhs`)
- `template<typename EXPR_VEC >`  
`Vector & operator+=` (const `VectorBase`< RTYPE, false, EXPR\_VEC > &`rhs`)
- bool `containsElementNamed` (const char \*`target`) const
- int `findName` (const std::string &`name`) const
- SEXP `eval` () const
- SEXP `eval` (SEXP `env`) const

## Static Public Member Functions

- static `stored_type get_na` ()
- static bool `is_na` (`stored_type` `x`)
- `template<typename InputIterator >`  
static `Vector import` (InputIterator `first`, InputIterator `last`)
- `template<typename InputIterator , typename F >`  
static `Vector import_transform` (InputIterator `first`, InputIterator `last`, F `f`)
- `template<typename U >`  
static void `replace_element` (`iterator` `it`, SEXP `names`, R\_xlen\_t `index`, const U &`u`)
- `template<typename U >`  
static void `replace_element__dispatch` (traits::false\_type, `iterator` `it`, SEXP `names`, R\_xlen\_t `index`, const U &`u`)
- `template<typename U >`  
static void `replace_element__dispatch` (traits::true\_type, `iterator` `it`, SEXP `names`, R\_xlen\_t `index`, const U &`u`)
- `template<typename U >`  
static void `replace_element__dispatch__isArgument` (traits::false\_type, `iterator` `it`, SEXP `names`, R\_xlen\_t `index`, const U &`u`)
- `template<typename U >`  
static void `replace_element__dispatch__isArgument` (traits::true\_type, `iterator` `it`, SEXP `names`, R\_xlen\_t `index`, const U &`u`)
- static `Vector create` ()

## Public Attributes

- `traits::r_vector_cache_type`< RTYPE, StoragePolicy >::type `cache`

## Protected Member Functions

- int \* `dims` () const
- void `init` ()

## Private Member Functions

- void `push_back__impl` (const `stored_type` &object, `traits::true_type`)
- void `push_back__impl` (const `stored_type` &object, `traits::false_type`)
- void `push_back_name__impl` (const `stored_type` &object, const std::string &name, `traits::true_type`)
- void `push_back_name__impl` (const `stored_type` &object, const std::string &name, `traits::false_type`)
- void `push_front__impl` (const `stored_type` &object, `traits::true_type`)
- void `push_front__impl` (const `stored_type` &object, `traits::false_type`)
- void `push_front_name__impl` (const `stored_type` &object, const std::string &name, `traits::true_type`)
- void `push_front_name__impl` (const `stored_type` &object, const std::string &name, `traits::false_type`)
- `iterator insert__impl` (`iterator` position, const `stored_type` &object\_, `traits::true_type`)
- `iterator insert__impl` (`iterator` position, const `stored_type` &object, `traits::false_type`)
- `iterator erase_single__impl` (`iterator` position)
- `iterator erase_range__impl` (`iterator` first, `iterator` last)
- template<typename T >  
void `assign_sugar_expression` (const T &x)
- template<typename T >  
void `assign_object` (const T &x, `traits::true_type`)
- template<typename T >  
void `assign_object` (const T &x, `traits::false_type`)
- template<bool NA, typename VEC >  
void `import_sugar_expression` (const `Rcpp::VectorBase`< RTYPE, NA, VEC > &other, `traits::false_type`)
- template<bool NA, typename VEC >  
void `import_sugar_expression` (const `Rcpp::VectorBase`< RTYPE, NA, VEC > &other, `traits::true_type`)
- template<typename T >  
void `import_expression` (const T &other, R\_xlen\_t n)
- template<typename T >  
void `fill_or_generate` (const T &t)
- template<typename T >  
void `fill_or_generate__impl` (const T &gen, `traits::true_type`)
- template<typename T >  
void `fill_or_generate__impl` (const T &t, `traits::false_type`)
- template<typename U >  
void `fill__dispatch` (`traits::false_type`, const U &u)
- template<typename U >  
void `fill__dispatch` (`traits::true_type`, const U &u)

### 6.813.1 Detailed Description

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
class Rcpp::Vector< RTYPE, StoragePolicy >
```

#### Examples

[ConvolveBenchmarks/convolve3\\_cpp.cpp](#), and [ConvolveBenchmarks/convolve4\\_cpp.cpp](#).

Definition at line 29 of file `Vector.h`.

## 6.813.2 Member Typedef Documentation

### 6.813.2.1 const\_iterator

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_const_iterator<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, StoragePolicy >::const_iterator
```

Definition at line 47 of file Vector.h.

### 6.813.2.2 const\_Proxy

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_const_proxy<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, StoragePolicy >::const_Proxy
```

Definition at line 43 of file Vector.h.

### 6.813.2.3 converter\_type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_element_converter<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >::converter_type
```

Definition at line 49 of file Vector.h.

### 6.813.2.4 Indexer

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef internal::RangeIndexer<RTYPE,true,Vector> Rcpp::Vector< RTYPE, StoragePolicy >::Indexer
```

Definition at line 545 of file Vector.h.

### 6.813.2.5 `init_type`

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::init_type<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >::init_type
```

Definition at line 48 of file Vector.h.

### 6.813.2.6 `iterator`

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_iterator<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, StoragePolicy
>::iterator
```

Definition at line 46 of file Vector.h.

### 6.813.2.7 `NameProxy`

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_name_proxy<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, Storage↵
Policy >::NameProxy
```

Definition at line 44 of file Vector.h.

### 6.813.2.8 `Proxy`

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_proxy<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, StoragePolicy
>::Proxy
```

Definition at line 42 of file Vector.h.

### 6.813.2.9 `Storage`

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef StoragePolicy<Vector> Rcpp::Vector< RTYPE, StoragePolicy >::Storage
```

Definition at line 39 of file Vector.h.

### 6.813.2.10 stored\_type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::storage_type<RTYPE>::type Rcpp::Vector< RTYPE, StoragePolicy >::stored_type
```

Definition at line 50 of file Vector.h.

### 6.813.2.11 value\_type

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
typedef traits::r_vector_proxy<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, StoragePolicy
>::value_type
```

Definition at line 45 of file Vector.h.

## 6.813.3 Constructor & Destructor Documentation

### 6.813.3.1 Vector() [1/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector ( ) [inline]
```

Default constructor. Creates a vector of the appropriate type and 0 length

Definition at line 56 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::init().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::create(), and Rcpp::Vector< RTYPE, StoragePolicy >::import\_transform().

Here is the call graph for this function:



**6.813.3.2 Vector()** [2/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const Vector< RTYPE, StoragePolicy > & other ) [inline]
```

copy constructor. shallow copy of the SEXP

Definition at line 64 of file Vector.h.

**6.813.3.3 Vector()** [3/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    SEXP x ) [inline]
```

Definition at line 72 of file Vector.h.

**6.813.3.4 Vector()** [4/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename Proxy >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const GenericProxy< Proxy > & proxy ) [inline]
```

Definition at line 78 of file Vector.h.

References Rcpp::GenericProxy< Proxy >::get().

Here is the call graph for this function:





## 6.813.3.5 Vector() [5/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const no_init_vector & obj ) [inline], [explicit]
```

Definition at line 83 of file Vector.h.

References Rcpp::no\_init\_vector::get().

Here is the call graph for this function:



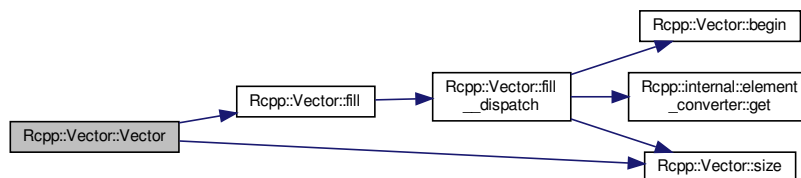
## 6.813.3.6 Vector() [6/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const T & size,
    const stored_type & u,
    typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 ) [inline]
```

Definition at line 88 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::fill(), RCPP\_DEBUG\_2, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



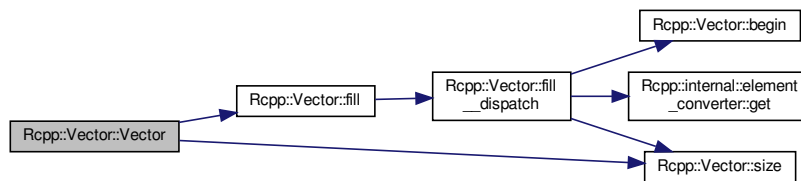
**6.813.3.7 Vector()** [7/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const int & size,
    const stored_type & u ) [inline]
```

Definition at line 95 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::fill(), RCPP\_DEBUG\_2, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:

**6.813.3.8 Vector()** [8/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const std::string & st ) [inline]
```

Definition at line 102 of file Vector.h.

References RCPP\_DEBUG\_2.

**6.813.3.9 Vector()** [9/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const char * st ) [inline]
```

Definition at line 108 of file Vector.h.

References RCPP\_DEBUG\_2.

## 6.813.3.10 Vector() [10/25]

```

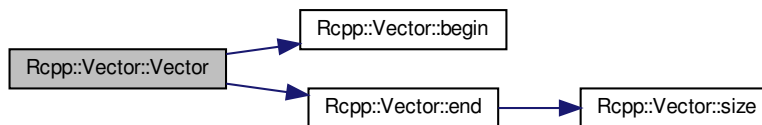
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const T & siz,
    stored_type(*) (void) gen,
    typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 ) [inline]

```

Definition at line 114 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), and RCPP\_DEBUG\_2.

Here is the call graph for this function:



## 6.813.3.11 Vector() [11/25]

```

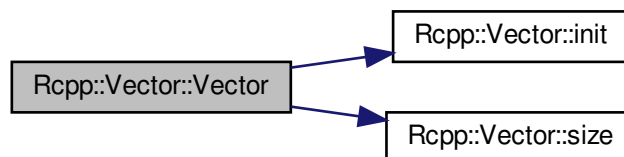
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    T size,
    typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 ) [inline]

```

Definition at line 123 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::init(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



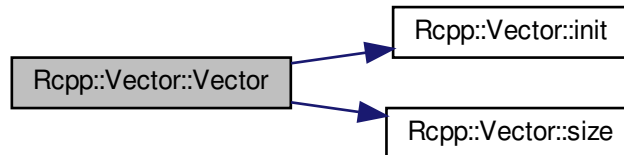
**6.813.3.12 Vector()** [12/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const int & size ) [inline]
```

Definition at line 129 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::init(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:

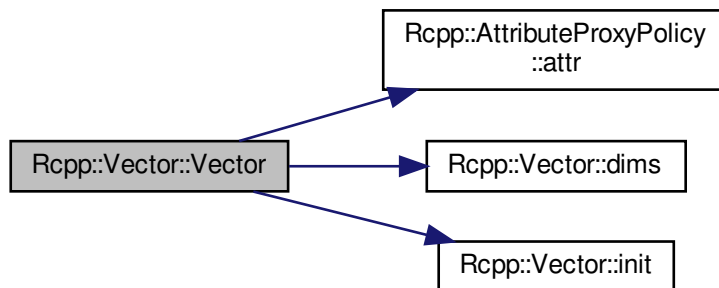
**6.813.3.13 Vector()** [13/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const Dimension & dims ) [inline]
```

Definition at line 134 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::dims(), and Rcpp::Vector< RTYPE, StoragePolicy >::init().

Here is the call graph for this function:



#### 6.813.3.14 Vector() [14/25]

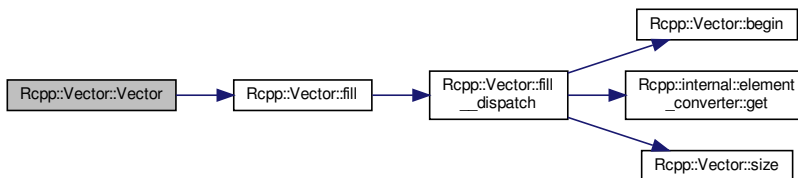
```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    T value,
    typename Rcpp::traits::enable_if< traits::is_bool< T >::value &&RTYPE==LGLSXP, void
>::type * = 0 ) [inline]
  
```

Definition at line 146 of file Vector.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::fill()`.

Here is the call graph for this function:



**6.813.3.15 Vector()** [15/25]

```

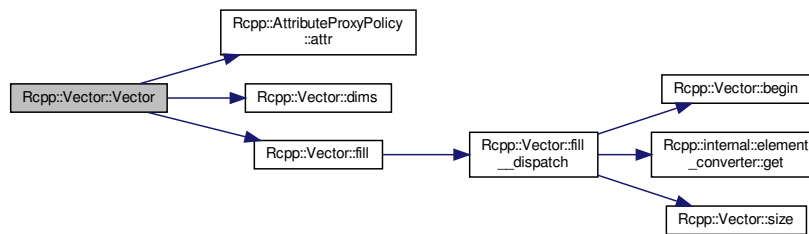
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const Dimension & dims,
    const U & u ) [inline]

```

Definition at line 153 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::dims(), Rcpp::Vector< RTYPE, StoragePolicy >::fill(), and RCPP\_DEBUG\_2.

Here is the call graph for this function:

**6.813.3.16 Vector()** [16/25]

```

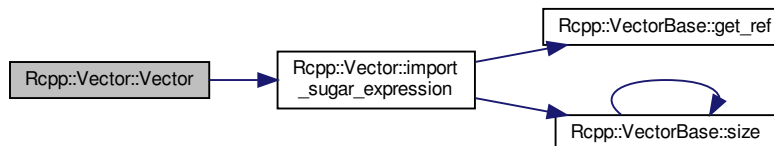
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<bool NA, typename VEC >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const VectorBase< RTYPE, NA, VEC > & other ) [inline]

```

Definition at line 163 of file Vector.h.

References DEMANGLE, Rcpp::Vector< RTYPE, StoragePolicy >::import\_sugar\_expression(), and RCPP\_DEBUG\_2.

Here is the call graph for this function:



## 6.813.3.17 Vector() [17/25]

```

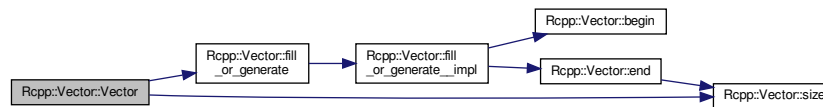
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T , typename U >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const T & size,
    const U & u,
    typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 ) [inline]

```

Definition at line 169 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::fill\_or\_generate(), RCPP\_DEBUG\_2, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



## 6.813.3.18 Vector() [18/25]

```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<bool NA, typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const sugar::SingleLogicalResult< NA, T > & obj ) [inline]

```

Definition at line 177 of file Vector.h.

References DEMANGLE, and RCPP\_DEBUG\_2.

**6.813.3.19 Vector()** [19/25]

```

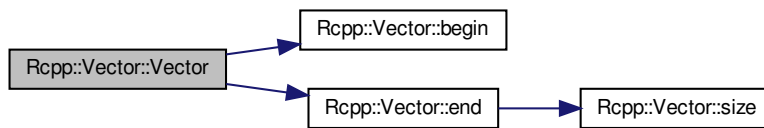
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T , typename U1 >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const T & siz,
    stored_type(*) (U1) gen,
    const U1 & u1,
    typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 ) [inline]

```

Definition at line 184 of file Vector.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, and `Rcpp::Vector::size`.

Here is the call graph for this function:

**6.813.3.20 Vector()** [20/25]

```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T , typename U1 , typename U2 >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const T & siz,
    stored_type(*) (U1, U2) gen,
    const U1 & u1,
    const U2 & u2,
    typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 ) [inline]

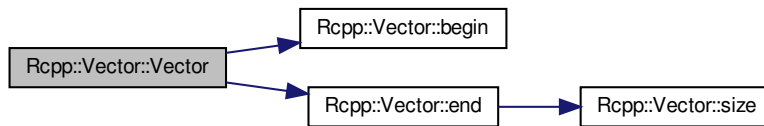
```

Definition at line 193 of file Vector.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, and `Rcpp::Vector::size`.



Here is the call graph for this function:



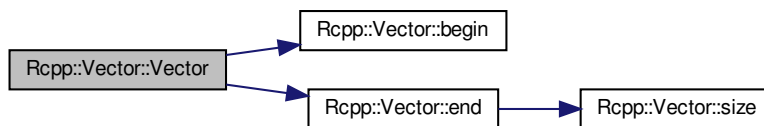
### 6.813.3.21 Vector() [21/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T , typename U1 , typename U2 , typename U3 >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    const T & siz,
    stored_type(*) (U1, U2, U3) gen,
    const U1 & u1,
    const U2 & u2,
    const U3 & u3,
    typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
= 0 ) [inline]
```

Definition at line 202 of file Vector.h.

References [Rcpp::Vector< RTYPE, StoragePolicy >::begin\(\)](#), [Rcpp::Vector< RTYPE, StoragePolicy >::end\(\)](#), and [RCPP\\_DEBUG\\_2](#).

Here is the call graph for this function:



**6.813.3.22 Vector()** [22/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    InputIterator first,
    InputIterator last ) [inline]
```

Definition at line 211 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and RCPP\_DEBUG\_1.

Here is the call graph for this function:

**6.813.3.23 Vector()** [23/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator , typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    InputIterator first,
    InputIterator last,
    T n,
    typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
    = 0 ) [inline]
```

Definition at line 218 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and RCPP\_DEBUG\_2.

Here is the call graph for this function:



**6.813.3.24 Vector()** [24/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator , typename Func >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    InputIterator first,
    InputIterator last,
    Func func ) [inline]
```

Definition at line 226 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and RCPP\_DEBUG\_1.

Here is the call graph for this function:

**6.813.3.25 Vector()** [25/25]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator , typename Func , typename T >
Rcpp::Vector< RTYPE, StoragePolicy >::Vector (
    InputIterator first,
    InputIterator last,
    Func func,
    T n,
    typename Rcpp::traits::enable_if< traits::is_arithmetic< T >::value, void >::type *
    = 0 ) [inline]
```

Definition at line 233 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and RCPP\_DEBUG\_2.

Here is the call graph for this function:



## 6.813.4 Member Function Documentation

### 6.813.4.1 assign()

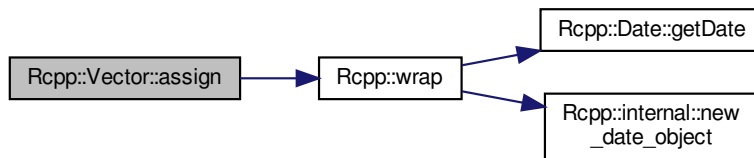
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator >
void Rcpp::Vector< RTYPE, StoragePolicy >::assign (
    InputIterator first,
    InputIterator last ) [inline]
```

Definition at line 430 of file Vector.h.

References Rcpp::wrap().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::import().

Here is the call graph for this function:



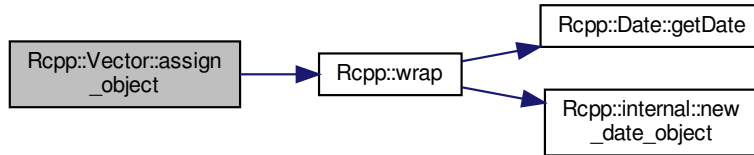
### 6.813.4.2 assign\_object() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::assign_object (
    const T & x,
    traits::false_type ) [inline], [private]
```

Definition at line 1059 of file Vector.h.

References Rcpp::wrap().

Here is the call graph for this function:



### 6.813.4.3 assign\_object() [2/2]

```

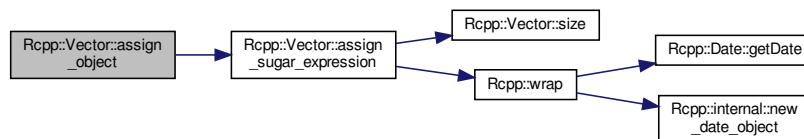
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::assign_object (
    const T & x,
    traits::true_type ) [inline], [private]
  
```

Definition at line 1053 of file Vector.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::assign_sugar_expression()`.

Referenced by `Rcpp::Vector< RTYPE, StoragePolicy >::operator=()`.

Here is the call graph for this function:



#### 6.813.4.4 `assign_sugar_expression()`

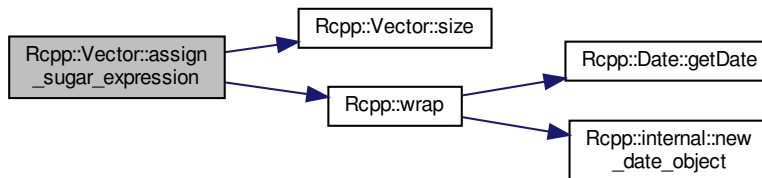
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::assign_sugar_expression (
    const T & x ) [inline], [private]
```

Definition at line 1038 of file Vector.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::size()`, and `Rcpp::wrap()`.

Referenced by `Rcpp::Vector< RTYPE, StoragePolicy >::assign_object()`.

Here is the call graph for this function:



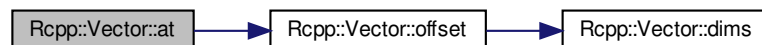
#### 6.813.4.5 `at()` [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Vector< RTYPE, StoragePolicy >::at (
    const size_t & i ) [inline]
```

Definition at line 351 of file Vector.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::cache`, and `Rcpp::Vector< RTYPE, StoragePolicy >::offset()`.

Here is the call graph for this function:



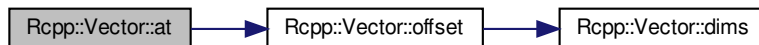
## 6.813.4.6 at() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_Proxy Rcpp::Vector< RTYPE, StoragePolicy >::at (
    const size_t & i ) const [inline]
```

Definition at line 354 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



## 6.813.4.7 begin() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::begin ( ) [inline]
```

## Examples

[ConvolveBenchmarks/convolve4\\_cpp.cpp](#).

Definition at line 334 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache.

Referenced by Rcpp::ListOf< T >::begin(), Rcpp::Matrix< RTYPE, StoragePolicy >::begin(), Rcpp::Matrix< RTYPE, StoragePolicy >::cbegin(), compileAttributes(), convolve10cpp(), convolve12cpp(), convolve4cpp(), convolve8cpp(), convolve9cpp(), Rcpp::exception::copy\_stack\_trace\_to\_r(), Rcpp::sugar::EmpiricalSample(), Rcpp::Vector< RTYPE, StoragePolicy >::erase\_range\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::erase\_single\_\_impl(), fastLm(), Rcpp::Vector< RTYPE, StoragePolicy >::fill\_\_dispatch(), Rcpp::Vector< RTYPE, StoragePolicy >::fill\_or\_generate\_\_impl(), Rcpp::sugar::Mean< RTYPE, NA, T >::get(), Rcpp::sugar::Mean< INTSXP, NA, T >::get(), Rcpp::sugar::SetDiff< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::get(), Rcpp::sugar::Intersect< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::get(), Rcpp::sugar::Union< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::get(), Rcpp::Vector< RTYPE, StoragePolicy >::import\_expression(), Rcpp::Matrix< RTYPE, StoragePolicy >::import\_matrix\_\_expression(), Rcpp::Vector< RTYPE, StoragePolicy >::insert\_\_impl(), lapplyCpp(), Rcpp::ListOf< T >::ListOf(), Rcpp::sugar::IndexHash< RTYPE >::lookup(), Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), Rcpp::Vector< RTYPE, StoragePolicy >::operator+=(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_back\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_back\_name\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_front\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_front\_name\_\_impl(), Rcpp::sugar::SampleNoReplace(), Rcpp::sugar::SampleReplace(), Rcpp::sugar::SelfMatch< RTYPE, TABLE\_T >::SelfMatch(), Rcpp::sugar::Table< RTYPE, TABLE\_T >::Table(), and Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

#### 6.813.4.8 begin() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>  
const_iterator Rcpp::Vector< RTYPE, StoragePolicy >::begin ( ) const [inline]
```

Definition at line 336 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache.

#### 6.813.4.9 cbegin()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>  
const_iterator Rcpp::Vector< RTYPE, StoragePolicy >::cbegin ( ) const [inline]
```

Definition at line 338 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache.

#### 6.813.4.10 cend()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>  
const_iterator Rcpp::Vector< RTYPE, StoragePolicy >::cend ( ) const [inline]
```

Definition at line 339 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:





## 6.813.4.11 containsElementNamed()

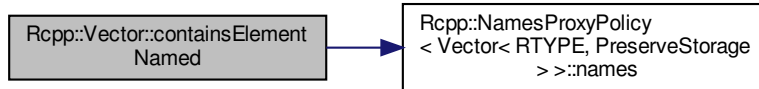
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
bool Rcpp::Vector< RTYPE, StoragePolicy >::containsElementNamed (
    const char * target ) const [inline]
```

Does this vector have an element with the target name

Definition at line 586 of file Vector.h.

References Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), and RCPP\_GET\_NAMES.

Here is the call graph for this function:



## 6.813.4.12 create()

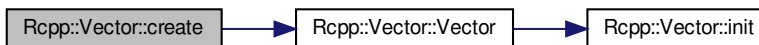
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
static Vector Rcpp::Vector< RTYPE, StoragePolicy >::create ( ) [inline], [static]
```

Definition at line 1122 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

Referenced by Rcpp::sugar::Range< RTYPE, NA, T >::operator Vector< RTYPE >(), Rcpp::sugar::Range< RTYPE, false, T >::operator Vector< RTYPE >(), Rcpp::shush\_about\_NA(), and Rcpp::trimws().

Here is the call graph for this function:



**6.813.4.13 dims()**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int* Rcpp::Vector< RTYPE, StoragePolicy >::dims ( ) const [inline], [protected]
```

Definition at line 613 of file Vector.h.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::cols(), Rcpp::Matrix< RTYPE, StoragePolicy >::ncol(), Rcpp::Vector< RTYPE, StoragePolicy >::offset(), and Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

**6.813.4.14 end() [1/2]**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::end ( ) [inline]
```

Definition at line 335 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::cend(), compileAttributes(), Rcpp::sugar::EmpiricalSample(), Rcpp::ListOf< T >::end(), Rcpp::Matrix< RTYPE, StoragePolicy >::end(), Rcpp::Vector< RTYPE, StoragePolicy >::erase\_range\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::erase\_single\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::fill\_dispatch(), Rcpp::Vector< RTYPE, StoragePolicy >::fill\_or\_generate\_\_impl(), Rcpp::sugar::Mean< RTYPE, NA, T >::get(), Rcpp::sugar::Mean< INTSXP, NA, T >::get(), Rcpp::Vector< RTYPE, StoragePolicy >::insert\_\_impl(), lapplyCpp(), Rcpp::ListOf< T >::ListOf(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_back\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_back\_name\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_front\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_front\_name\_\_impl(), Rcpp::sugar::SelfMatch< RTYPE, TABLE\_T >::SelfMatch(), Rcpp::sugar::Table< RTYPE, TABLE\_T >::Table(), and Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

Here is the call graph for this function:



**6.813.4.15 end()** [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_iterator Rcpp::Vector< RTYPE, StoragePolicy >::end ( ) const [inline]
```

Definition at line 337 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:

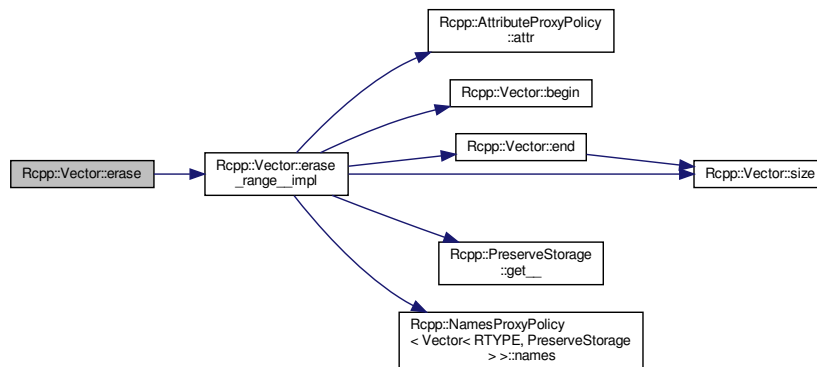
**6.813.4.16 erase()** [1/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase (
    int first,
    int last ) [inline]
```

Definition at line 500 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::erase\_range\_impl().

Here is the call graph for this function:



**6.813.4.17 erase()** [2/4]

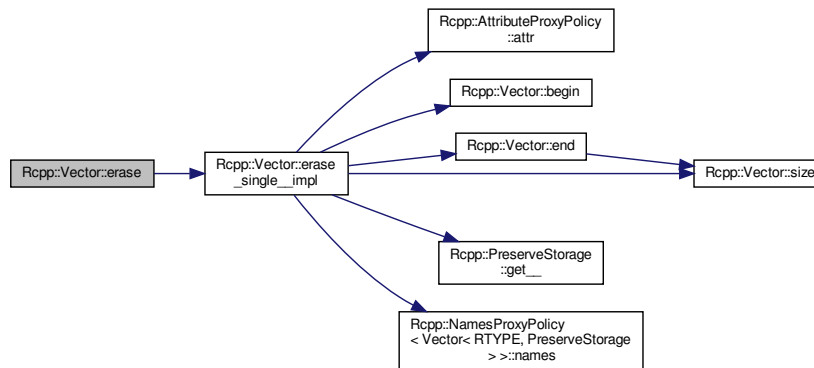
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase (
    int position ) [inline]
```

Definition at line 492 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::erase\_←single\_\_impl().

Referenced by Rcpp::DataFrame\_Impl< StoragePolicy >::from\_list().

Here is the call graph for this function:

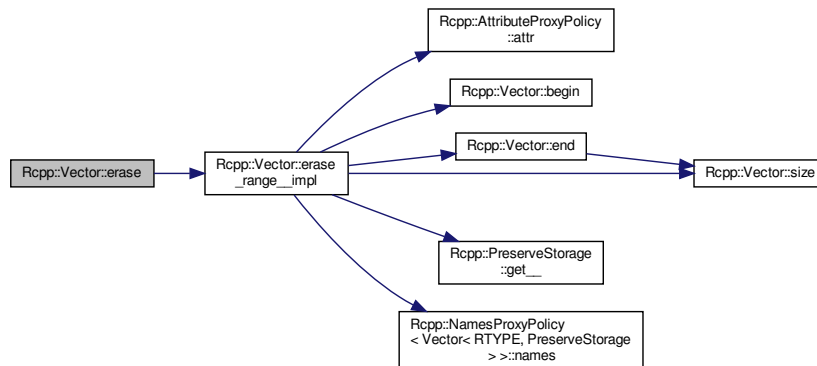
**6.813.4.18 erase()** [3/4]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase (
    iterator first,
    iterator last ) [inline]
```

Definition at line 505 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::erase\_range\_\_impl().

Here is the call graph for this function:



#### 6.813.4.19 erase() [4/4]

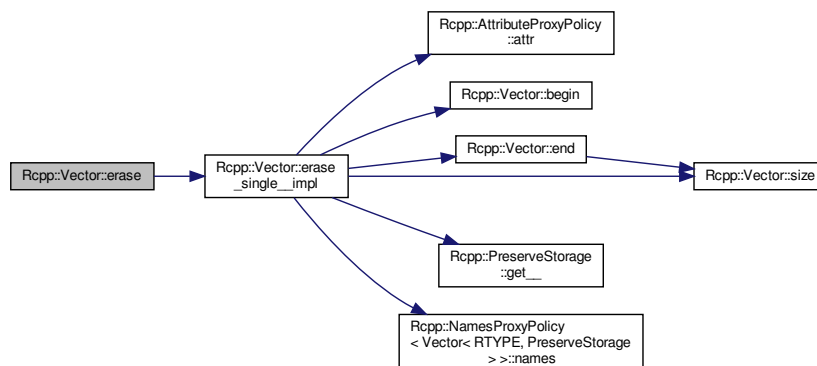
```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase (
    iterator position ) [inline]
  
```

Definition at line 496 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::erase\_single\_impl().

Here is the call graph for this function:



### 6.813.4.20 erase\_range\_impl()

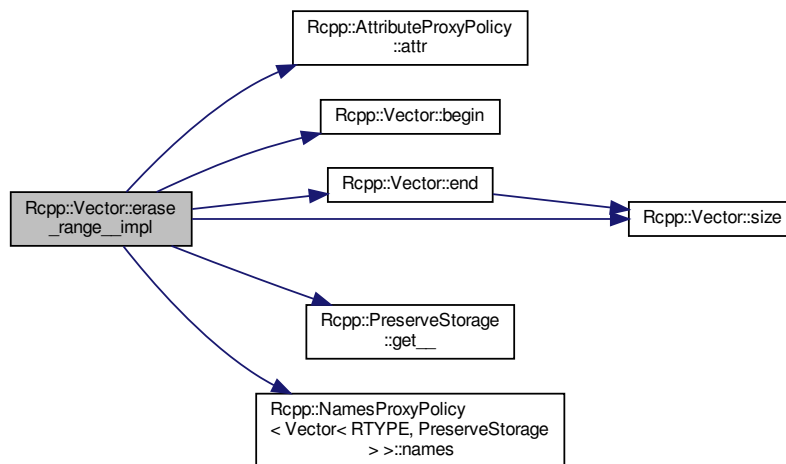
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase_range_impl (
    iterator first,
    iterator last ) [inline], [private]
```

Definition at line 978 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get\_\_(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP\_GET\_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::erase().

Here is the call graph for this function:



### 6.813.4.21 erase\_single\_impl()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::erase_single_impl (
    iterator position ) [inline], [private]
```

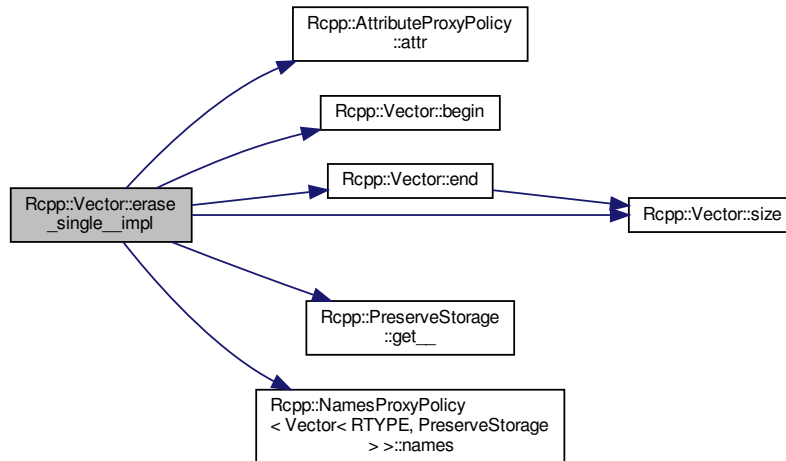
Definition at line 924 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get\_\_(), Rcpp::NamesProxyPolicy<

Vector< RTYPE, PreserveStorage > >::names(), RCPP\_GET\_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::erase().

Here is the call graph for this function:



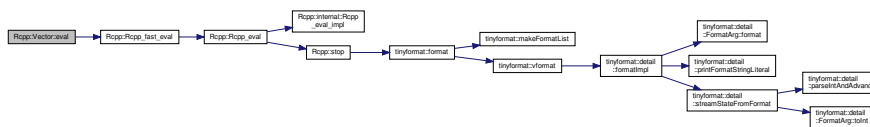
#### 6.813.4.22 eval() [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
SEXP Rcpp::Vector< RTYPE, StoragePolicy >::eval ( ) const [inline]
```

Definition at line 1130 of file Vector.h.

References `Rcpp::Rcpp_fast_eval()`.

Here is the call graph for this function:



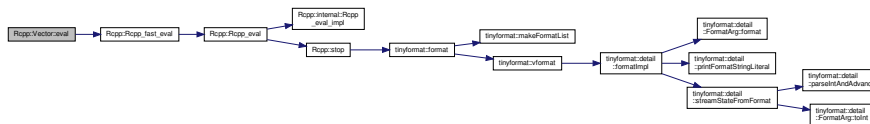
### 6.813.4.23 eval() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
SEXP Rcpp::Vector< RTYPE, StoragePolicy >::eval (
    SEXP env ) const [inline]
```

Definition at line 1134 of file Vector.h.

References Rcpp::Rcpp\_fast\_eval().

Here is the call graph for this function:



### 6.813.4.24 fill()

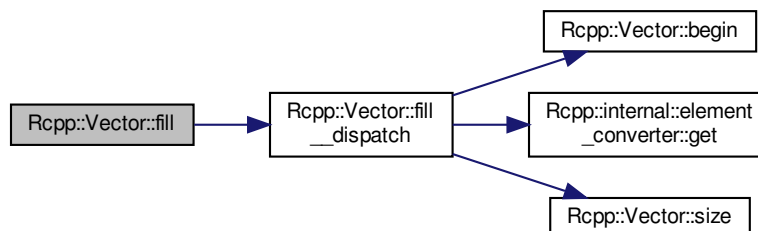
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
void Rcpp::Vector< RTYPE, StoragePolicy >::fill (
    const U & u ) [inline]
```

Definition at line 330 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::fill\_\_dispatch().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::fill\_or\_generate\_\_impl(), and Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

Here is the call graph for this function:





**6.813.4.25 fill\_\_dispatch()** [1/2]

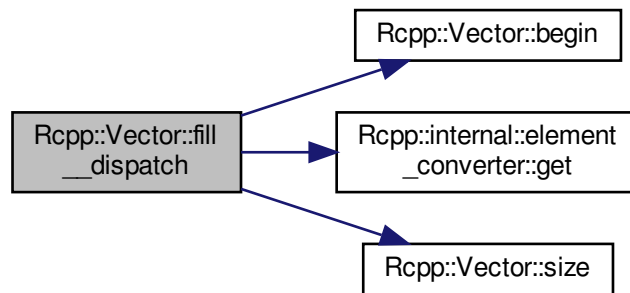
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
void Rcpp::Vector< RTYPE, StoragePolicy >::fill__dispatch (
    traits::false_type ,
    const U & u ) [inline], [private]
```

Definition at line 1106 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::internal::element\_converter< RTYPE >::get(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::fill().

Here is the call graph for this function:

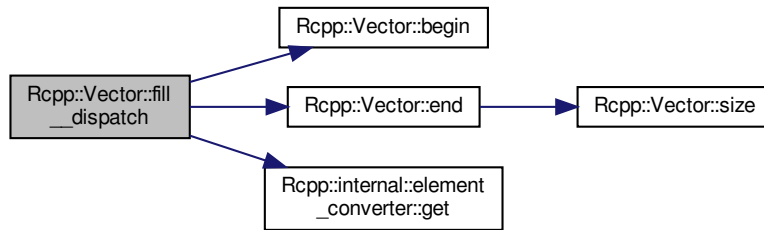
**6.813.4.26 fill\_\_dispatch()** [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
void Rcpp::Vector< RTYPE, StoragePolicy >::fill__dispatch (
    traits::true_type ,
    const U & u ) [inline], [private]
```

Definition at line 1116 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), and Rcpp::internal::element\_converter< RTYPE >::get().

Here is the call graph for this function:



#### 6.813.4.27 fill\_or\_generate()

```

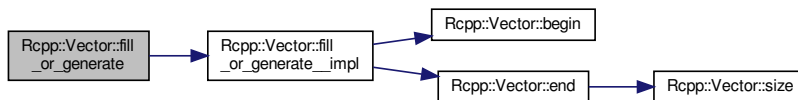
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::fill_or_generate (
    const T & t ) [inline], [private]
  
```

Definition at line 1089 of file Vector.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::fill_or_generate__impl()`.

Referenced by `Rcpp::Vector< RTYPE, StoragePolicy >::Vector()`.

Here is the call graph for this function:



## 6.813.4.28 fill\_or\_generate\_\_impl() [1/2]

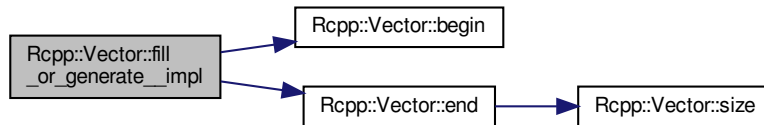
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::fill_or_generate__impl (
    const T & gen,
    traits::true_type ) [inline], [private]
```

Definition at line 1094 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and Rcpp::Vector< RTYPE, StoragePolicy >::end().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::fill\_or\_generate().

Here is the call graph for this function:



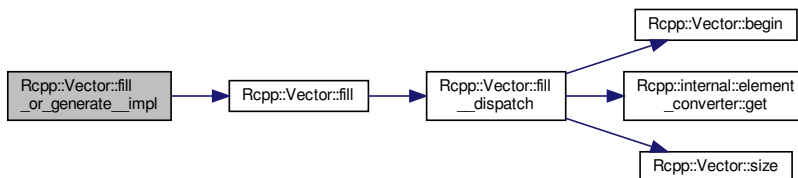
## 6.813.4.29 fill\_or\_generate\_\_impl() [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::fill_or_generate__impl (
    const T & t,
    traits::false_type ) [inline], [private]
```

Definition at line 1101 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::fill().

Here is the call graph for this function:



### 6.813.4.30 findName()

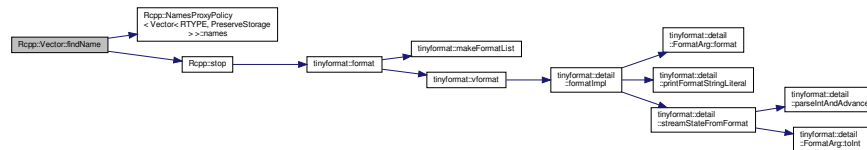
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
int Rcpp::Vector< RTYPE, StoragePolicy >::findName (
    const std::string & name ) const [inline]
```

Definition at line 597 of file Vector.h.

References Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP\_GET\_NAMES, and Rcpp::stop().

Referenced by Rcpp::ListOf< T >::operator[]().

Here is the call graph for this function:



### 6.813.4.31 get\_na()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
static stored_type Rcpp::Vector< RTYPE, StoragePolicy >::get_na ( ) [inline], [static]
```

Definition at line 252 of file Vector.h.

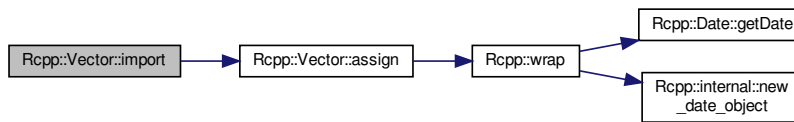
### 6.813.4.32 import()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator >
static Vector Rcpp::Vector< RTYPE, StoragePolicy >::import (
    InputIterator first,
    InputIterator last ) [inline], [static]
```

Definition at line 440 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::assign().

Here is the call graph for this function:



#### 6.813.4.33 import\_expression()

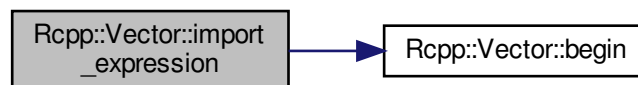
```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::import_expression (
    const T & other,
    R_xlen_t n ) [inline], [private]
  
```

Definition at line 1083 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), and RCPP\_LOOP\_UNROLL.

Here is the call graph for this function:



#### 6.813.4.34 import\_sugar\_expression() [1/2]

```

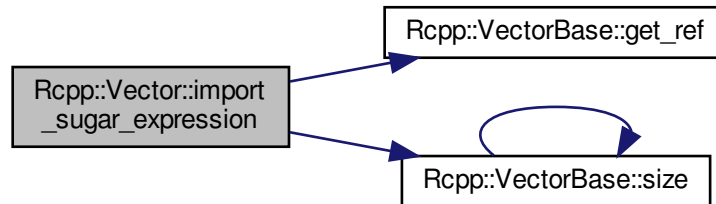
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<bool NA, typename VEC >
void Rcpp::Vector< RTYPE, StoragePolicy >::import_sugar_expression (
    const Rcpp::VectorBase< RTYPE, NA, VEC > & other,
    traits::false_type ) [inline], [private]
  
```

Definition at line 1067 of file Vector.h.

References DEMANGLE, Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref(), Rcpp::NA, RCPP\_DEBUG\_4, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

Here is the call graph for this function:



#### 6.813.4.35 import\_sugar\_expression() [2/2]

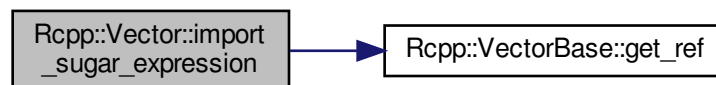
```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<bool NA, typename VEC >
void Rcpp::Vector< RTYPE, StoragePolicy >::import_sugar_expression (
    const Rcpp::VectorBase< RTYPE, NA, VEC > & other,
    traits::true_type ) [inline], [private]
  
```

Definition at line 1076 of file Vector.h.

References DEMANGLE, Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref(), Rcpp::NA, and RCPP\_DEBUG\_4.

Here is the call graph for this function:



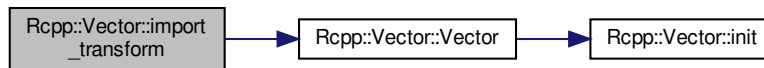
#### 6.813.4.36 import\_transform()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename InputIterator , typename F >
static Vector Rcpp::Vector< RTYPE, StoragePolicy >::import_transform (
    InputIterator first,
    InputIterator last,
    F f ) [inline], [static]
```

Definition at line 447 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::Vector().

Here is the call graph for this function:



#### 6.813.4.37 init()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::init ( ) [inline], [protected]
```

Definition at line 617 of file Vector.h.

References RCPP\_DEBUG\_2.

Referenced by Rcpp::Matrix< RTYPE, StoragePolicy >::Matrix(), and Rcpp::Vector< RTYPE, StoragePolicy >::← Vector().

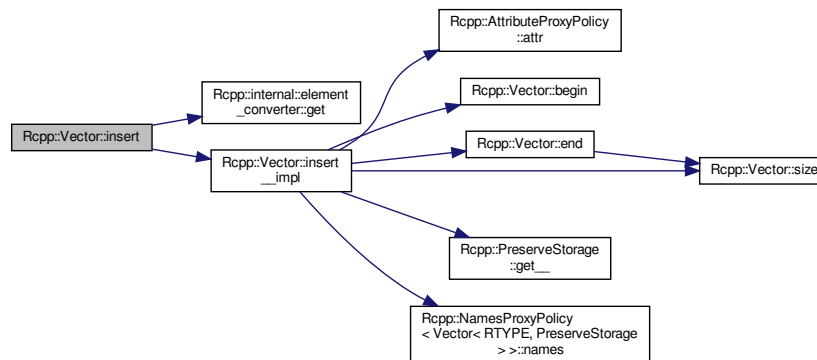
**6.813.4.38 insert()** [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
iterator Rcpp::Vector< RTYPE, StoragePolicy >::insert (
    int position,
    const T & object ) [inline]
```

Definition at line 486 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, Rcpp::internal::element\_converter< RTYPE >::get(), and Rcpp::Vector< RTYPE, StoragePolicy >::insert\_\_impl().

Here is the call graph for this function:

**6.813.4.39 insert()** [2/2]

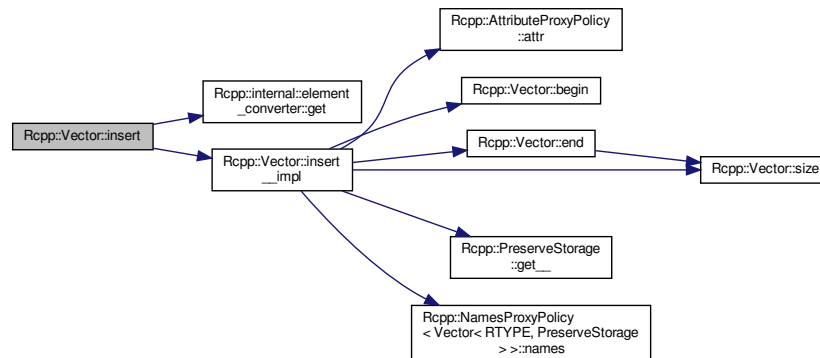
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
iterator Rcpp::Vector< RTYPE, StoragePolicy >::insert (
    iterator position,
    const T & object ) [inline]
```

Definition at line 479 of file Vector.h.

References Rcpp::internal::element\_converter< RTYPE >::get(), and Rcpp::Vector< RTYPE, StoragePolicy >::insert\_\_impl().



Here is the call graph for this function:



#### 6.813.4.40 insert\_\_impl() [1/2]

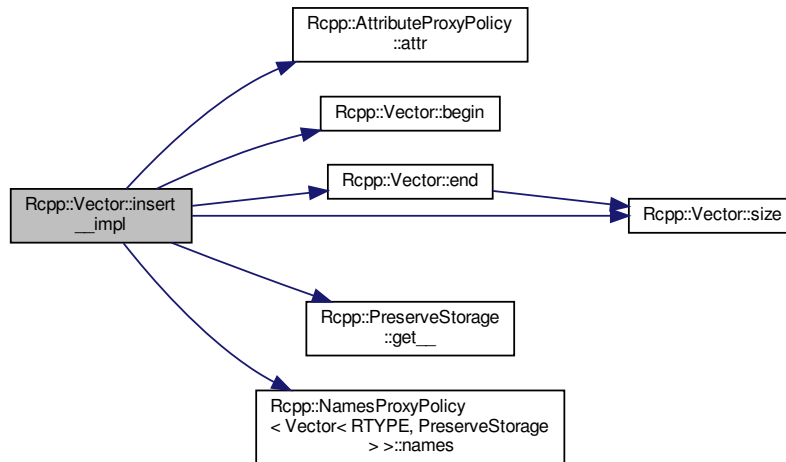
```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::insert__impl (
    iterator position,
    const stored_type & object,
    traits::false_type ) [inline], [private]
  
```

Definition at line 884 of file Vector.h.

References `Rcpp::AttributeProxyPolicy< CLASS >::attr()`, `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::PreserveStorage< CLASS >::get__()`, `Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names()`, `RCPP_GET_NAMES`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



#### 6.813.4.41 `insert__impl()` [2/2]

```

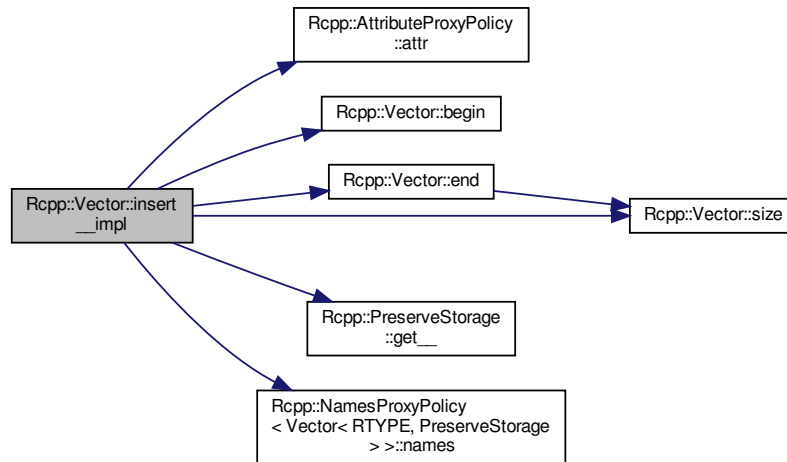
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
iterator Rcpp::Vector< RTYPE, StoragePolicy >::insert__impl (
    iterator position,
    const stored_type & object_,
    traits::true_type ) [inline], [private]
  
```

Definition at line 843 of file `Vector.h`.

References `Rcpp::AttributeProxyPolicy< CLASS >::attr()`, `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::PreserveStorage< CLASS >::get__()`, `Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names()`, `RCPP_GET_NAMES`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Referenced by `Rcpp::Vector< RTYPE, StoragePolicy >::insert()`.

Here is the call graph for this function:



#### 6.813.4.42 is\_na()

```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
static bool Rcpp::Vector< RTYPE, StoragePolicy >::is_na (
    stored_type x ) [inline], [static]
  
```

Definition at line 255 of file Vector.h.

#### 6.813.4.43 length()

```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
R_xlen_t Rcpp::Vector< RTYPE, StoragePolicy >::length ( ) const [inline]
  
```

the length of the vector, uses Rf\_xlength

Definition at line 269 of file Vector.h.

**6.813.4.44 offset()** [1/3]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
R_xlen_t Rcpp::Vector< RTYPE, StoragePolicy >::offset (
    const int & i,
    const int & j ) const [inline]
```

offset based on the dimensions of this vector

Definition at line 283 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::dims().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::at(), Rcpp::internal::simple\_name\_proxy< RTYPE, StoragePolicy >::get(), Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::get(), Rcpp::internal::generic\_name\_proxy< RTYPE, StoragePolicy >::get(), Rcpp::Vector< RTYPE, StoragePolicy >::operator(), Rcpp::internal::string\_name\_proxy< RTYPE, StoragePolicy >::set(), Rcpp::internal::simple\_name\_proxy< RTYPE, StoragePolicy >::set(), and Rcpp::internal::generic\_name\_proxy< RTYPE, StoragePolicy >::set().

Here is the call graph for this function:

**6.813.4.45 offset()** [2/3]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
R_xlen_t Rcpp::Vector< RTYPE, StoragePolicy >::offset (
    const R_xlen_t & i ) const [inline]
```

one dimensional offset doing bounds checking to ensure it is valid

Definition at line 303 of file Vector.h.

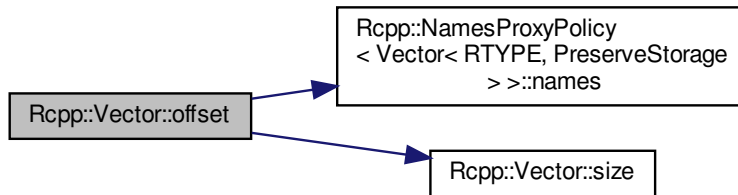
**6.813.4.46 offset()** [3/3]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
R_xlen_t Rcpp::Vector< RTYPE, StoragePolicy >::offset (
    const std::string & name ) const [inline]
```

Definition at line 311 of file Vector.h.

References Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP\_GET\_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:

**6.813.4.47 operator RObject()**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Rcpp::Vector< RTYPE, StoragePolicy >::operator RObject ( ) const [inline]
```

Definition at line 379 of file Vector.h.

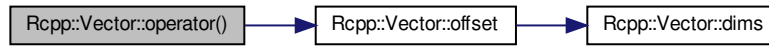
**6.813.4.48 operator>()** [1/6]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (
    const size_t & i ) [inline]
```

Definition at line 344 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:



#### 6.813.4.49 operator>() [2/6]

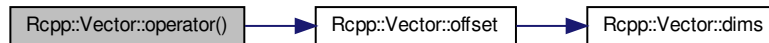
```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (
    const size_t & i ) const [inline]
  
```

Definition at line 347 of file Vector.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::cache`, and `Rcpp::Vector< RTYPE, StoragePolicy >::offset()`.

Here is the call graph for this function:



#### 6.813.4.50 operator>() [3/6]

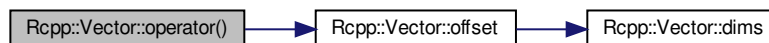
```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (
    const size_t & i,
    const size_t & j ) [inline]
  
```

Definition at line 358 of file Vector.h.

References `Rcpp::Vector< RTYPE, StoragePolicy >::cache`, and `Rcpp::Vector< RTYPE, StoragePolicy >::offset()`.

Here is the call graph for this function:



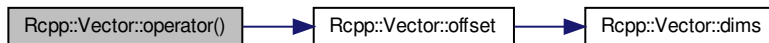
**6.813.4.51 operator>() [4/6]**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (
    const size_t & i,
    const size_t & j ) const [inline]
```

Definition at line 361 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache, and Rcpp::Vector< RTYPE, StoragePolicy >::offset().

Here is the call graph for this function:

**6.813.4.52 operator>() [5/6]**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
NameProxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (
    const std::string & name ) [inline]
```

Definition at line 368 of file Vector.h.

**6.813.4.53 operator>() [6/6]**

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
NameProxy Rcpp::Vector< RTYPE, StoragePolicy >::operator() (
    const std::string & name ) const [inline]
```

Definition at line 375 of file Vector.h.

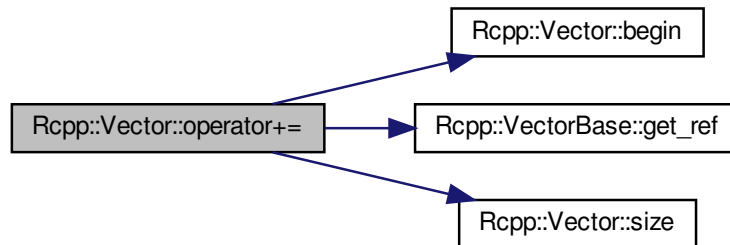
**6.813.4.54 operator+=()** [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename EXPR_VEC >
Vector& Rcpp::Vector< RTYPE, StoragePolicy >::operator+= (
    const VectorBase< RTYPE, false, EXPR_VEC > & rhs ) [inline]
```

Definition at line 569 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:

**6.813.4.55 operator+=()** [2/2]

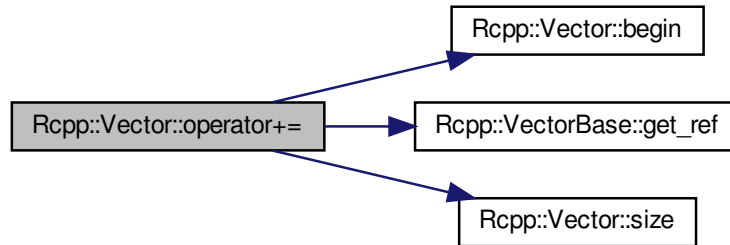
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename EXPR_VEC >
Vector& Rcpp::Vector< RTYPE, StoragePolicy >::operator+= (
    const VectorBase< RTYPE, true, EXPR_VEC > & rhs ) [inline]
```

Definition at line 552 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::VectorBase< RTYPE, na, VECTOR >::get\_ref(), and Rcpp::Vector< RTYPE, StoragePolicy >::size().



Here is the call graph for this function:



#### 6.813.4.56 operator=() [1/2]

```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
Vector& Rcpp::Vector< RTYPE, StoragePolicy >::operator= (
    const T & x ) [inline]
  
```

Definition at line 247 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::assign\_object().

Here is the call graph for this function:



#### 6.813.4.57 operator=() [2/2]

```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Vector& Rcpp::Vector< RTYPE, StoragePolicy >::operator= (
    const Vector< RTYPE, StoragePolicy > & rhs ) [inline]
  
```

Definition at line 68 of file Vector.h.

Referenced by Rcpp::newDatetimeVector::operator=(), and Rcpp::newDateVector::operator=().

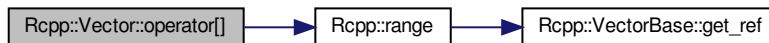
**6.813.4.58 operator[]()** [1/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Indexer Rcpp::Vector< RTYPE, StoragePolicy >::operator[] (
    const Range & range ) [inline]
```

Definition at line 547 of file Vector.h.

References Rcpp::range().

Here is the call graph for this function:

**6.813.4.59 operator[]()** [2/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
NameProxy Rcpp::Vector< RTYPE, StoragePolicy >::operator[] (
    const std::string & name ) [inline]
```

Definition at line 365 of file Vector.h.

**6.813.4.60 operator[]()** [3/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
NameProxy Rcpp::Vector< RTYPE, StoragePolicy >::operator[] (
    const std::string & name ) const [inline]
```

Definition at line 372 of file Vector.h.

**6.813.4.61 operator[]()** [4/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<int RHS_RTYPE, bool RHS_NA, typename RHS_T >
SubsetProxy<RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T> Rcpp::Vector< RTYPE, StoragePolicy
>::operator[] (
    const VectorBase< RHS_RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 386 of file Vector.h.

**6.813.4.62 operator[]()** [5/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<int RHS_RTYPE, bool RHS_NA, typename RHS_T >
const SubsetProxy<RTYPE, StoragePolicy, RHS_RTYPE, RHS_NA, RHS_T> Rcpp::Vector< RTYPE, StoragePolicy >::operator[] (
    const VectorBase< RHS_RTYPE, RHS_NA, RHS_T > & rhs ) const [inline]
```

Definition at line 395 of file Vector.h.

**6.813.4.63 operator[]()** [6/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator[] (
    R_xlen_t i ) [inline]
```

Definition at line 341 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache.

**6.813.4.64 operator[]()** [7/7]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
const_Proxy Rcpp::Vector< RTYPE, StoragePolicy >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 342 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache.

### 6.813.4.65 `push_back()` [1/2]

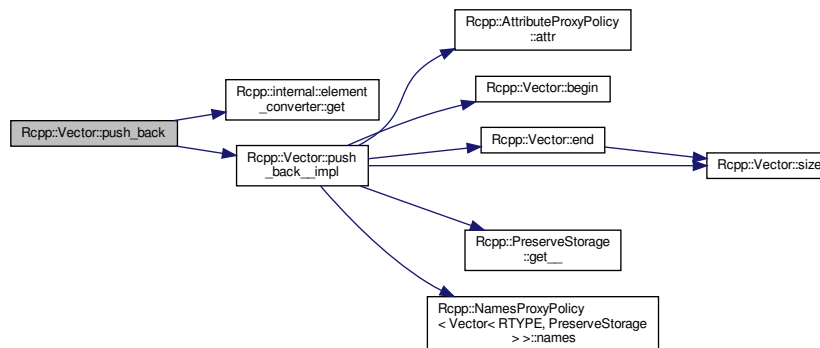
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::push_back (
    const T & object ) [inline]
```

Definition at line 452 of file Vector.h.

References `Rcpp::internal::element_converter< RTYPE >::get()`, and `Rcpp::Vector< RTYPE, StoragePolicy >::push_back_impl()`.

Referenced by `Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >::set()`, `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >::set()`, `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >::set()`, and `Rcpp::attributes::CppExportsGenerator::writeEnd()`.

Here is the call graph for this function:



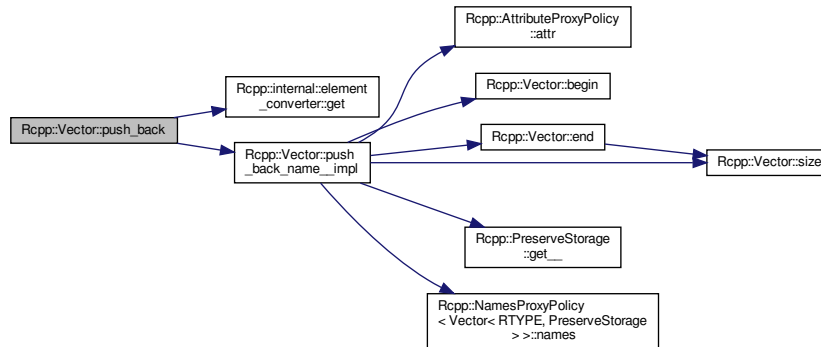
### 6.813.4.66 `push_back()` [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::push_back (
    const T & object,
    const std::string & name ) [inline]
```

Definition at line 459 of file Vector.h.

References `Rcpp::internal::element_converter< RTYPE >::get()`, and `Rcpp::Vector< RTYPE, StoragePolicy >::push_back_name_impl()`.

Here is the call graph for this function:



#### 6.813.4.67 push\_back\_\_impl() [1/2]

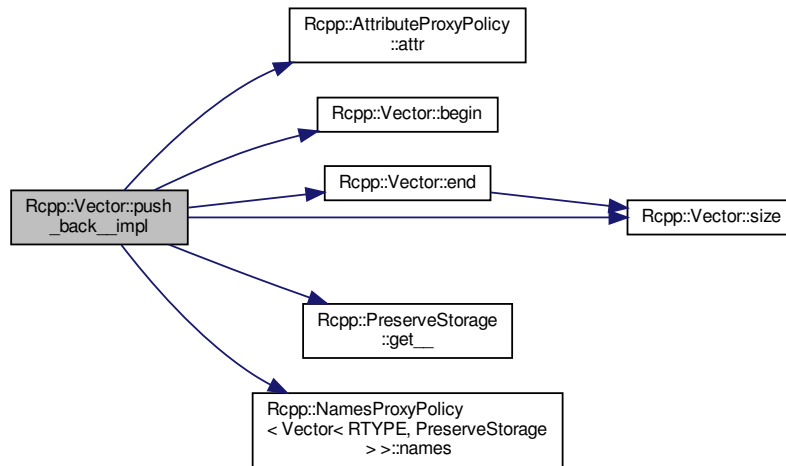
```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_back__impl (
    const stored_type & object,
    traits::false_type ) [inline], [private]
  
```

Definition at line 650 of file Vector.h.

References `Rcpp::AttributeProxyPolicy< CLASS >::attr()`, `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::PreserveStorage< CLASS >::get_()`, `Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names()`, `RCPP_GET_NAMES`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



#### 6.813.4.68 `push_back_impl()` [2/2]

```

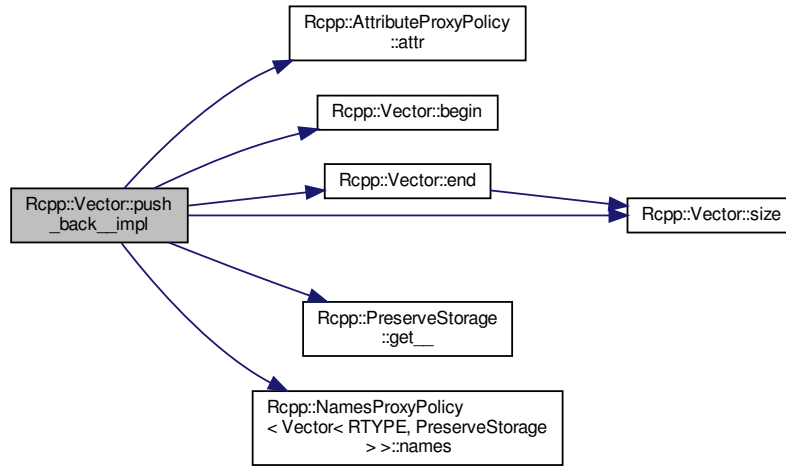
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_back_impl (
    const stored_type & object,
    traits::true_type ) [inline], [private]
  
```

Definition at line 624 of file `Vector.h`.

References `Rcpp::AttributeProxyPolicy< CLASS >::attr()`, `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::PreserveStorage< CLASS >::get_()`, `Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names()`, `Rcpp_GET_NAMES`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Referenced by `Rcpp::Vector< RTYPE, StoragePolicy >::push_back()`.

Here is the call graph for this function:



#### 6.813.4.69 push\_back\_name\_\_impl() [1/2]

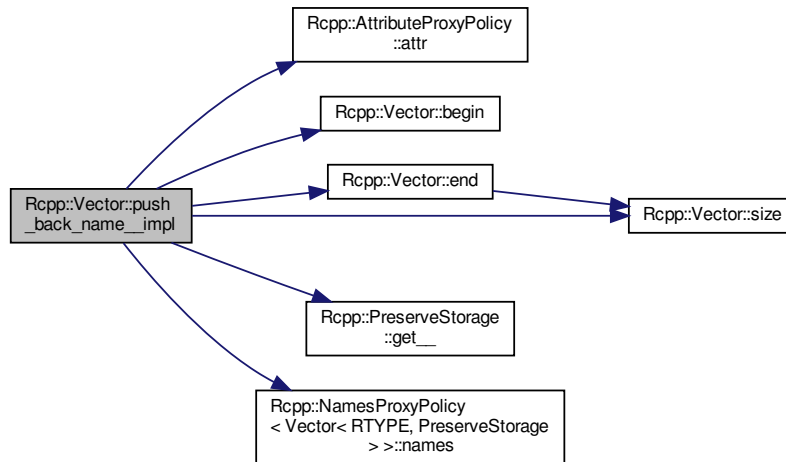
```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_back_name__impl (
    const stored_type & object,
    const std::string & name,
    traits::false_type ) [inline], [private]
  
```

Definition at line 702 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get\_\_(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP\_GET\_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



#### 6.813.4.70 push\_back\_name\_impl() [2/2]

```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_back_name_impl (
    const stored_type & object,
    const std::string & name,
    traits::true_type ) [inline], [private]
  
```

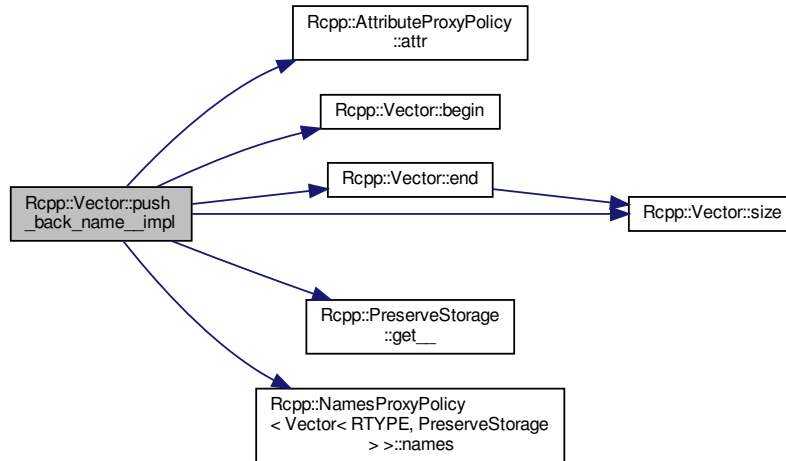
Definition at line 675 of file Vector.h.

References `Rcpp::AttributeProxyPolicy< CLASS >::attr()`, `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::PreserveStorage< CLASS >::get_()`, `Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names()`, `RCPP_GET_NAMES`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Referenced by `Rcpp::Vector< RTYPE, StoragePolicy >::push_back()`.



Here is the call graph for this function:



#### 6.813.4.71 push\_front() [1/2]

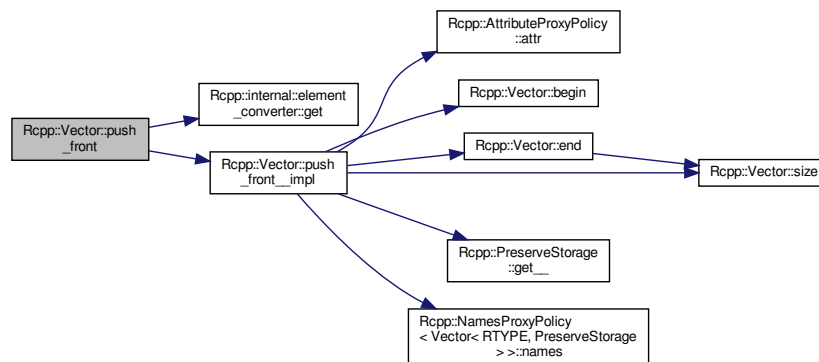
```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::push_front (
    const T & object ) [inline]
  
```

Definition at line 466 of file Vector.h.

References Rcpp::internal::element\_converter< RTYPE >::get(), and Rcpp::Vector< RTYPE, StoragePolicy >::push←\_front\_impl().

Here is the call graph for this function:



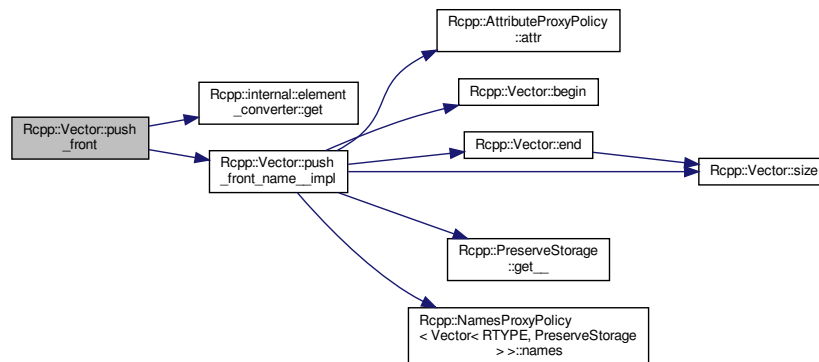
### 6.813.4.72 `push_front()` [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename T >
void Rcpp::Vector< RTYPE, StoragePolicy >::push_front (
    const T & object,
    const std::string & name ) [inline]
```

Definition at line 472 of file Vector.h.

References `Rcpp::internal::element_converter< RTYPE >::get()`, and `Rcpp::Vector< RTYPE, StoragePolicy >::push_front_name_impl()`.

Here is the call graph for this function:



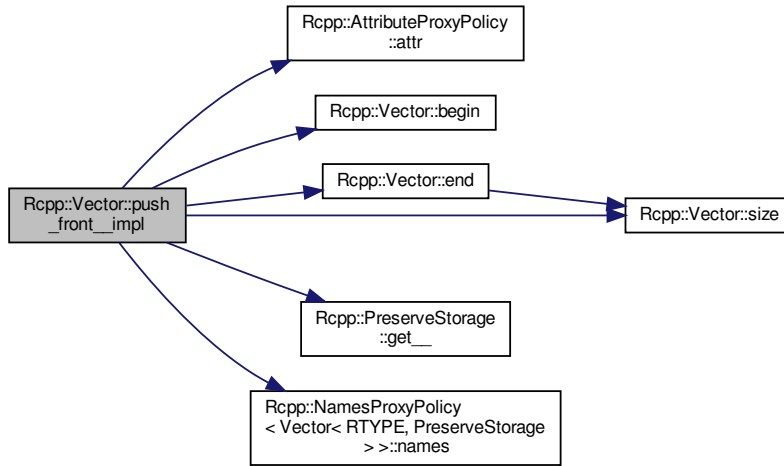
### 6.813.4.73 `push_front_impl()` [1/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_front_impl (
    const stored_type & object,
    traits::false_type ) [inline], [private]
```

Definition at line 757 of file Vector.h.

References `Rcpp::AttributeProxyPolicy< CLASS >::attr()`, `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::PreserveStorage< CLASS >::get_()`, `Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names()`, `Rcpp_GET_NAMES`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



#### 6.813.4.74 push\_front\_impl() [2/2]

```

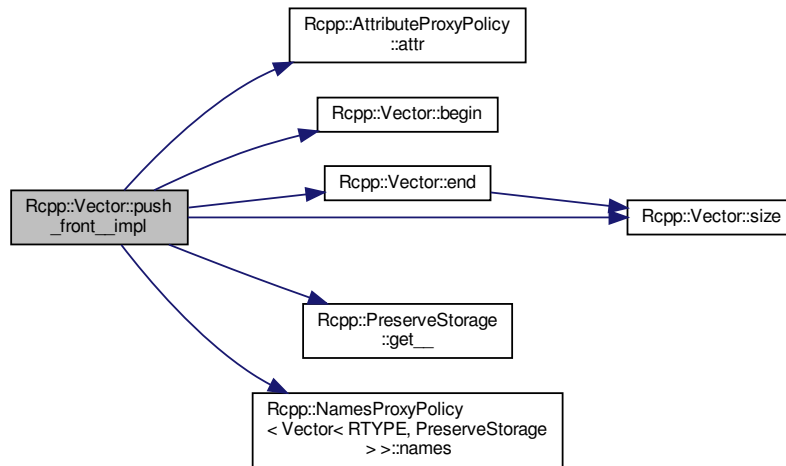
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_front_impl (
    const stored_type & object,
    traits::true_type ) [inline], [private]
  
```

Definition at line 730 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get\_(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP\_GET\_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::push\_front().

Here is the call graph for this function:



#### 6.813.4.75 push\_front\_name\_\_impl() [1/2]

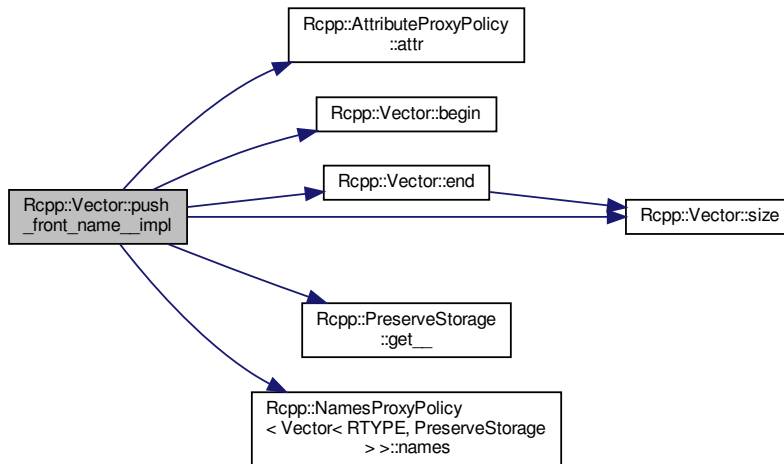
```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_front_name__impl (
    const stored_type & object,
    const std::string & name,
    traits::false_type ) [inline], [private]
  
```

Definition at line 813 of file Vector.h.

References `Rcpp::AttributeProxyPolicy< CLASS >::attr()`, `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::PreserveStorage< CLASS >::get_()`, `Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names()`, `RCPP_GET_NAMES`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



#### 6.813.4.76 push\_front\_name\_\_impl() [2/2]

```

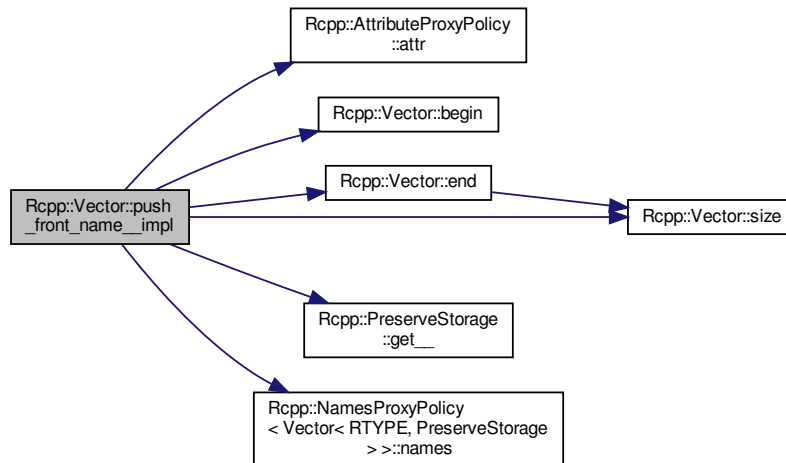
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::push_front_name__impl (
    const stored_type & object,
    const std::string & name,
    traits::true_type ) [inline], [private]
  
```

Definition at line 784 of file Vector.h.

References Rcpp::AttributeProxyPolicy< CLASS >::attr(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp::Vector< RTYPE, StoragePolicy >::end(), Rcpp::PreserveStorage< CLASS >::get\_\_(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), RCPP\_GET\_NAMES, and Rcpp::Vector< RTYPE, StoragePolicy >::size().

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::push\_front().

Here is the call graph for this function:



#### 6.813.4.77 replace\_element()

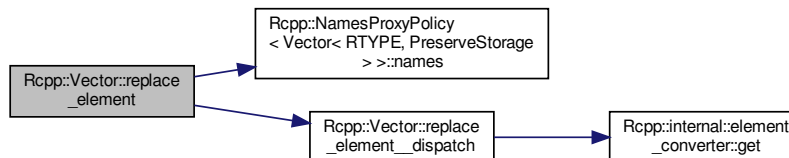
```

template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
static void Rcpp::Vector< RTYPE, StoragePolicy >::replace_element (
    iterator it,
    SEXP names,
    R_xlen_t index,
    const U & u ) [inline], [static]
  
```

Definition at line 514 of file `Vector.h`.

References `Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names()`, and `Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch()`.

Here is the call graph for this function:



6.813.4.78 `replace_element__dispatch()` [1/2]

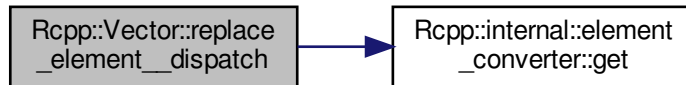
```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
static void Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch (
    traits::false_type ,
    iterator it,
    SEXP names,
    R_xlen_t index,
    const U & u ) [inline], [static]
```

Definition at line 520 of file Vector.h.

References `Rcpp::internal::element_converter< RTYPE >::get()`.

Referenced by `Rcpp::Vector< RTYPE, StoragePolicy >::replace_element()`.

Here is the call graph for this function:

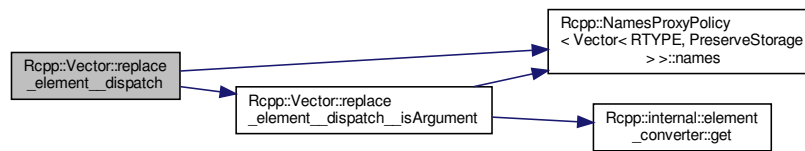
6.813.4.79 `replace_element__dispatch()` [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
static void Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch (
    traits::true_type ,
    iterator it,
    SEXP names,
    R_xlen_t index,
    const U & u ) [inline], [static]
```

Definition at line 525 of file Vector.h.

References `Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names()`, and `Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch__isArgument()`.

Here is the call graph for this function:



#### 6.813.4.80 replace\_element\_dispatch\_isArgument() [1/2]

```

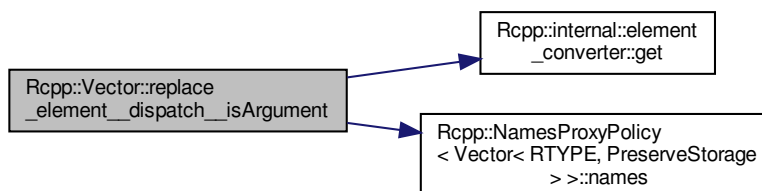
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
static void Rcpp::Vector< RTYPE, StoragePolicy >::replace_element_dispatch_isArgument (
    traits::false_type ,
    iterator it,
    SEXP names,
    R_xlen_t index,
    const U & u ) [inline], [static]
  
```

Definition at line 530 of file Vector.h.

References DEMANGLE, Rcpp::internal::element\_converter< RTYPE >::get(), Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names(), and RCPP\_DEBUG\_2.

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::replace\_element\_dispatch().

Here is the call graph for this function:





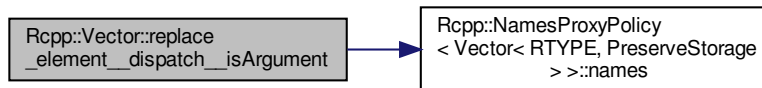
6.813.4.81 `replace_element__dispatch__isArgument()` [2/2]

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
template<typename U >
static void Rcpp::Vector< RTYPE, StoragePolicy >::replace_element__dispatch__isArgument (
    traits::true_type ,
    iterator it,
    SEXP names,
    R_xlen_t index,
    const U & u ) [inline], [static]
```

Definition at line 538 of file Vector.h.

References `DEMANGLE`, `Rcpp::NamesProxyPolicy< Vector< RTYPE, PreserveStorage > >::names()`, and `Rcpp::DEBUG_2`.

Here is the call graph for this function:

6.813.4.82 `size()`

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
R_xlen_t Rcpp::Vector< RTYPE, StoragePolicy >::size ( ) const [inline]
```

alias of `length`

## Examples

[ConvolveBenchmarks/convolve3\\_cpp.cpp](#), and [ConvolveBenchmarks/convolve4\\_cpp.cpp](#).

Definition at line 276 of file Vector.h.

Referenced by `Rcpp::Vector< RTYPE, StoragePolicy >::assign_sugar_expression()`, `Rcpp::Vector< RTYPE, StoragePolicy >::cend()`, `convolve()`, `convolve10cpp()`, `convolve11cpp()`, `convolve12cpp()`, `convolve14cpp()`, `convolve3cpp()`, `convolve4cpp()`, `convolve5cpp()`, `convolve8cpp()`, `convolve9cpp()`, `Rcpp::sugar::EmpiricalSample()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::Vector< RTYPE, StoragePolicy >::erase_range__impl()`, `Rcpp::Vector< RTYPE, StoragePolicy >::erase_single__impl()`, `fastLm()`, `Rcpp::Vector< RTYPE, StoragePolicy >::fill__dispatch()`, `Rcpp::DataFrame_Impl< StoragePolicy >::from_list()`, `Rcpp::sugar::Cummax< RTYPE, NA, T >::get()`, `Rcpp::sugar::Cummin< RTYPE, NA, T >::get()`, `Rcpp::sugar::Cumprod< RTYPE, NA, T >::get()`, `Rcpp::sugar::Cumsum< RTYPE,`

NA, T >::get(), Rcpp::sugar::Mean< RTYPE, NA, T >::get(), Rcpp::sugar::Mean< CPLXSXP, NA, T >::get(), Rcpp::sugar::Mean< LGLSXP, NA, T >::get(), Rcpp::sugar::Mean< INTSXP, NA, T >::get(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::get\_indices(), Rcpp::newDateVector::getDates(), Rcpp::newDatetimeVector::getDatetimes(), Rcpp::Vector< RTYPE, StoragePolicy >::insert\_\_impl(), lapplyCpp(), Rcpp::sugar::IndexHash< RTYPE >::lookup(), Rcpp::sugar::Normalize(), Rcpp::Vector< RTYPE, StoragePolicy >::offset(), Rcpp::Vector< RTYPE, StoragePolicy >::operator+=(), Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\_RTYPE, RHS\_NA, RHS\_T >::operator=(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_back\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_back\_name\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_front\_\_impl(), Rcpp::Vector< RTYPE, StoragePolicy >::push\_front\_name\_\_impl(), Rcpp::sample(), Rcpp::sugar::SampleNoReplace(), Rcpp::sugar::SampleReplace(), Rcpp::ListOf< T >::size(), Rcpp::Vector< RTYPE, StoragePolicy >::sort(), Rcpp::trimws(), Rcpp::Vector< RTYPE, StoragePolicy >::Vector(), and Rcpp::sugar::WalkerSample().

#### 6.813.4.83 sort()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
Vector& Rcpp::Vector< RTYPE, StoragePolicy >::sort (
    bool decreasing = false ) [inline]
```

Definition at line 402 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



#### 6.813.4.84 update()

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
void Rcpp::Vector< RTYPE, StoragePolicy >::update (
    SEXP ) [inline]
```

Definition at line 509 of file Vector.h.

References Rcpp::Vector< RTYPE, StoragePolicy >::cache.

## 6.813.5 Member Data Documentation

### 6.813.5.1 cache

```
template<int RTYPE, template< class > class StoragePolicy = PreserveStorage>
traits::r_vector_cache_type<RTYPE, StoragePolicy>::type Rcpp::Vector< RTYPE, StoragePolicy >↔
::cache
```

Definition at line 41 of file Vector.h.

Referenced by Rcpp::Vector< RTYPE, StoragePolicy >::at(), Rcpp::Vector< RTYPE, StoragePolicy >::begin(), Rcpp↔  
::Vector< RTYPE, StoragePolicy >::cbegin(), Rcpp::Vector< RTYPE, StoragePolicy >::cend(), Rcpp::Vector< RTYPE,  
StoragePolicy >::end(), Rcpp::Vector< RTYPE, StoragePolicy >::erase(), Rcpp::Vector< RTYPE, StoragePolicy >↔  
::insert(), Rcpp::Vector< RTYPE, StoragePolicy >::operator>(), Rcpp::Vector< RTYPE, StoragePolicy >::operator[](),  
and Rcpp::Vector< RTYPE, StoragePolicy >::update().

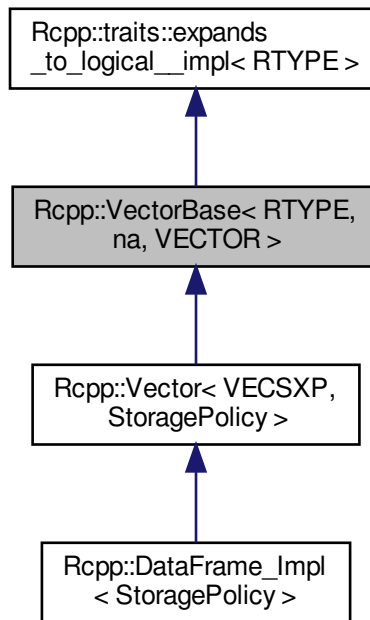
The documentation for this class was generated from the following files:

- [inst/include/Rcpp/vector/00\\_forward\\_Vector.h](#)
- [inst/include/Rcpp/vector/Vector.h](#)

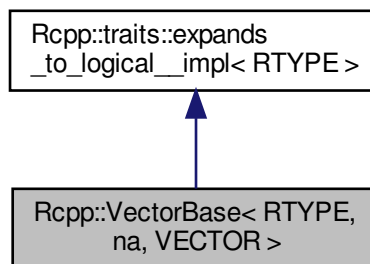
## 6.814 Rcpp::VectorBase< RTYPE, na, VECTOR > Class Template Reference

```
#include <VectorBase.h>
```

Inheritance diagram for `Rcpp::VectorBase< RTYPE, na, VECTOR >`:



Collaboration diagram for `Rcpp::VectorBase< RTYPE, na, VECTOR >`:



## Classes

- struct [can\\_have\\_na](#)

- struct [const\\_iter\\_traits](#)
- class [iter\\_base](#)
- struct [iter\\_traits](#)
- struct [r\\_type](#)
- struct [rcpp\\_sugar\\_expression](#)

## Public Types

- typedef [traits::storage\\_type](#)< RTYPE >::type [stored\\_type](#)
- typedef [traits::storage\\_type](#)< RTYPE >::type [elem\\_type](#)
- typedef [iter\\_base](#)< [iter\\_traits](#) > [iterator](#)
- typedef [iter\\_base](#)< [const\\_iter\\_traits](#) > [const\\_iterator](#)

## Public Member Functions

- VECTOR & [get\\_ref](#) ()
- const VECTOR & [get\\_ref](#) () const
- [stored\\_type operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const
- [const\\_iterator begin](#) () const
- [const\\_iterator end](#) () const
- [const\\_iterator cbegin](#) () const
- [const\\_iterator cend](#) () const

### 6.814.1 Detailed Description

```
template<int RTYPE, bool na, typename VECTOR>
class Rcpp::VectorBase< RTYPE, na, VECTOR >
```

a base class for vectors, modelled after the CRTP

Definition at line 29 of file VectorBase.h.

### 6.814.2 Member Typedef Documentation

#### 6.814.2.1 const\_iterator

```
template<int RTYPE, bool na, typename VECTOR >
typedef iter\_base< const\_iter\_traits > Rcpp::VectorBase< RTYPE, na, VECTOR >::const_iterator
```

Definition at line 162 of file VectorBase.h.

### 6.814.2.2 elem\_type

```
template<int RTYPE, bool na, typename VECTOR >  
typedef traits::storage\_type<RTYPE>::type Rcpp::VectorBase< RTYPE, na, VECTOR >::elem_type
```

Definition at line 35 of file VectorBase.h.

### 6.814.2.3 iterator

```
template<int RTYPE, bool na, typename VECTOR >  
typedef iter\_base< iter\_traits > Rcpp::VectorBase< RTYPE, na, VECTOR >::iterator
```

Definition at line 161 of file VectorBase.h.

### 6.814.2.4 stored\_type

```
template<int RTYPE, bool na, typename VECTOR >  
typedef traits::storage\_type<RTYPE>::type Rcpp::VectorBase< RTYPE, na, VECTOR >::stored_type
```

Definition at line 34 of file VectorBase.h.

## 6.814.3 Member Function Documentation

### 6.814.3.1 begin()

```
template<int RTYPE, bool na, typename VECTOR >  
const\_iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::begin ( ) const [inline]
```

Definition at line 164 of file VectorBase.h.

### 6.814.3.2 cbegin()

```
template<int RTYPE, bool na, typename VECTOR >  
const\_iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::cbegin ( ) const [inline]
```

Definition at line 167 of file VectorBase.h.

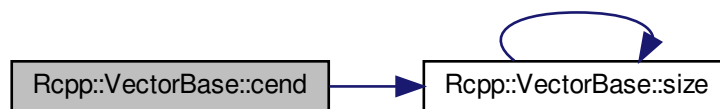
### 6.814.3.3 cend()

```
template<int RTYPE, bool na, typename VECTOR >  
const_iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::cend ( ) const [inline]
```

Definition at line 168 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



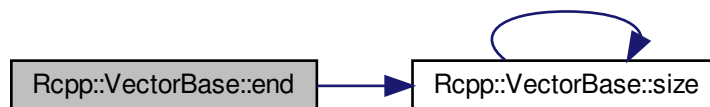
### 6.814.3.4 end()

```
template<int RTYPE, bool na, typename VECTOR >  
const_iterator Rcpp::VectorBase< RTYPE, na, VECTOR >::end ( ) const [inline]
```

Definition at line 165 of file VectorBase.h.

References Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



**6.814.3.5 get\_ref()** [1/2]

```
template<int RTYPE, bool na, typename VECTOR >
VECTOR& Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref ( ) [inline]
```

Definition at line 37 of file VectorBase.h.

Referenced by Rcpp::clamp(), Rcpp::collapse(), Rcpp::Vector< RTYPE, StoragePolicy >::import\_sugar\_expression(), Rcpp::in(), Rcpp::intersect(), Rcpp::is\_finite(), Rcpp::is\_infinite(), Rcpp::is\_nan(), Rcpp::mapply(), Rcpp::match(), Rcpp::max(), Rcpp::min(), Rcpp::na\_omit(), Rcpp::Vector< RTYPE, StoragePolicy >::operator+=(), Rcpp::Matrix↔Column< RTYPE >::operator=(), Rcpp::MatrixRow< RTYPE >::operator=(), Rcpp::pmax(), Rcpp::pmin(), Rcpp::pow(), Rcpp::range(), Rcpp::setdiff(), Rcpp::setequal(), Rcpp::table(), and Rcpp::union\_().

**6.814.3.6 get\_ref()** [2/2]

```
template<int RTYPE, bool na, typename VECTOR >
const VECTOR& Rcpp::VectorBase< RTYPE, na, VECTOR >::get_ref ( ) const [inline]
```

Definition at line 41 of file VectorBase.h.

**6.814.3.7 operator[]()**

```
template<int RTYPE, bool na, typename VECTOR >
stored_type Rcpp::VectorBase< RTYPE, na, VECTOR >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 45 of file VectorBase.h.

**6.814.3.8 size()**

```
template<int RTYPE, bool na, typename VECTOR >
R_xlen_t Rcpp::VectorBase< RTYPE, na, VECTOR >::size ( ) const [inline]
```

Definition at line 49 of file VectorBase.h.

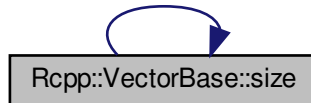
References Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Referenced by Rcpp::VectorBase< RTYPE, na, VECTOR >::end(), Rcpp::collapse(), Rcpp::VectorBase< RTYPE, na, VECTOR >::end(), Rcpp::sugar::WhichMax< RTYPE, NA, T >::get(), Rcpp::sugar::WhichMax< RTYPE, false, T >::get(), Rcpp::sugar::WhichMin< RTYPE, NA, T >::get(), Rcpp::sugar::WhichMin< RTYPE, false, T >::get(), Rcpp↔::Vector< RTYPE, StoragePolicy >::import\_sugar\_expression(), Rcpp::seq\_along(), Rcpp::stats::D0< RTYPE, NA, T >::size(), Rcpp::stats::D1< RTYPE, NA, T >::size(), Rcpp::stats::D2< RTYPE, NA, T >::size(), Rcpp::stats::↔D3< RTYPE, NA, T >::size(), Rcpp::stats::P0< RTYPE, NA, T >::size(), Rcpp::stats::P1< RTYPE, NA, T >::size(),



Rcpp::stats::P2< RTYPE, NA, T >::size(), Rcpp::stats::P3< RTYPE, NA, T >::size(), Rcpp::stats::Q0< RTYPE, NA, T >::size(), Rcpp::stats::Q1< RTYPE, NA, T >::size(), Rcpp::stats::Q2< RTYPE, NA, T >::size(), Rcpp::stats::Q3< RTYPE, NA, T >::size(), Rcpp::sugar::SugarMath\_1< NA, RESULT\_TYPE, U1, T1, FunPtr >::size(), Rcpp::sugar::SugarMath\_1< NA, RESULT\_TYPE, int, T1, FunPtr >::size(), Rcpp::sugar::SugarMath\_1< false, RESULT\_TYPE, int, T1, FunPtr >::size(), Rcpp::sugar::SugarComplex< NA, RESULT\_TYPE, T, FunPtr >::size(), Rcpp::sugar::Diff< RTYPE, LHS\_NA, LHS\_T >::size(), Rcpp::sugar::Diff< REALSXP, LHS\_NA, LHS\_T >::size(), Rcpp::sugar::Diff< RTYPE, false, LHS\_T >::size(), Rcpp::sugar::IfElse< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::IfElse< RTYPE, false, COND\_T, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, COND\_NA, COND\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::IfElse\_Primitive\_Vector< RTYPE, false, COND\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, COND\_NA, COND\_T, LHS\_NA, LHS\_T >::size(), Rcpp::sugar::IfElse\_Vector\_Primitive< RTYPE, false, COND\_T, LHS\_NA, LHS\_T >::size(), Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, COND\_NA, COND\_T >::size(), Rcpp::sugar::IfElse\_Primitive\_Primitive< RTYPE, false, COND\_T >::size(), Rcpp::sugar::IsNa< RTYPE, NA, VEC\_TYPE >::size(), Rcpp::sugar::IsNa< RTYPE, false, VEC\_TYPE >::size(), Rcpp::sugar::Lapply< RTYPE, NA, T, Function >::size(), Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< false, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, false, RHS\_T >::size(), Rcpp::sugar::And\_LogicalExpression\_LogicalExpression< false, LHS\_T, false, RHS\_T >::size(), Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< LHS\_NA, LHS\_T, false, RHS\_T >::size(), Rcpp::sugar::Or\_LogicalExpression\_LogicalExpression< false, LHS\_T, false, RHS\_T >::size(), Rcpp::sugar::Comparator< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, RHS\_NA, RHS\_T >::size(), Rcpp::sugar::Comparator< RTYPE, Operator, false, LHS\_T, false, RHS\_T >::size(), Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, NA, T >::size(), Rcpp::sugar::Comparator\_With\_One\_Value< RTYPE, Operator, false, T >::size(), Rcpp::sugar::Not\_Vector< RTYPE, NA, T >::size(), Rcpp::sugar::UnaryMinus\_Vector< RTYPE, NA, T >::size(), and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Here is the call graph for this function:



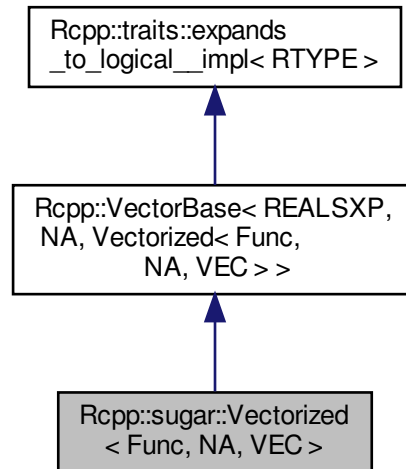
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/vector/VectorBase.h](#)

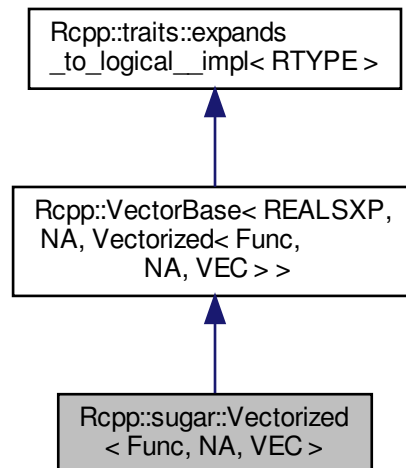
## 6.815 Rcpp::sugar::Vectorized< Func, NA, VEC > Class Template Reference

```
#include <Vectorized_Math.h>
```

Inheritance diagram for `Rcpp::sugar::Vectorized< Func, NA, VEC >`:



Collaboration diagram for `Rcpp::sugar::Vectorized< Func, NA, VEC >`:



## Public Types

- typedef `Rcpp::VectorBase< REALSXP, NA, VEC >` `VEC_TYPE`

- typedef [Rcpp::traits::Extractor](#)< REALSXP, NA, VEC >::type [VEC\\_EXT](#)

## Public Member Functions

- [Vectorized](#) (const [VEC\\_TYPE](#) &object\_)
- double [operator\[\]](#) (R\_xlen\_t i) const
- R\_xlen\_t [size](#) () const

## Private Attributes

- const [VEC\\_EXT](#) & [object](#)

### 6.815.1 Detailed Description

```
template<DDFun Func, bool NA, typename VEC>
class Rcpp::sugar::Vectorized< Func, NA, VEC >
```

Definition at line 31 of file `Vectorized_Math.h`.

### 6.815.2 Member Typedef Documentation

#### 6.815.2.1 VEC\_EXT

```
template<DDFun Func, bool NA, typename VEC >
typedef Rcpp::traits::Extractor<REALSXP, NA, VEC>::type Rcpp::sugar::Vectorized< Func, NA, VEC >↔
::VEC_EXT
```

Definition at line 34 of file `Vectorized_Math.h`.

#### 6.815.2.2 VEC\_TYPE

```
template<DDFun Func, bool NA, typename VEC >
typedef Rcpp::VectorBase<REALSXP, NA, VEC> Rcpp::sugar::Vectorized< Func, NA, VEC >::VEC_TYPE
```

Definition at line 33 of file `Vectorized_Math.h`.

### 6.815.3 Constructor & Destructor Documentation

### 6.815.3.1 Vectorized()

```
template<DDFun Func, bool NA, typename VEC >
Rcpp::sugar::Vectorized< Func, NA, VEC >::Vectorized (
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 36 of file Vectorized\_Math.h.

## 6.815.4 Member Function Documentation

### 6.815.4.1 operator[]()

```
template<DDFun Func, bool NA, typename VEC >
double Rcpp::sugar::Vectorized< Func, NA, VEC >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 37 of file Vectorized\_Math.h.

### 6.815.4.2 size()

```
template<DDFun Func, bool NA, typename VEC >
R_xlen_t Rcpp::sugar::Vectorized< Func, NA, VEC >::size ( ) const [inline]
```

Definition at line 40 of file Vectorized\_Math.h.

## 6.815.5 Member Data Documentation

### 6.815.5.1 object

```
template<DDFun Func, bool NA, typename VEC >
const VEC_EXT& Rcpp::sugar::Vectorized< Func, NA, VEC >::object [private]
```

Definition at line 43 of file Vectorized\_Math.h.

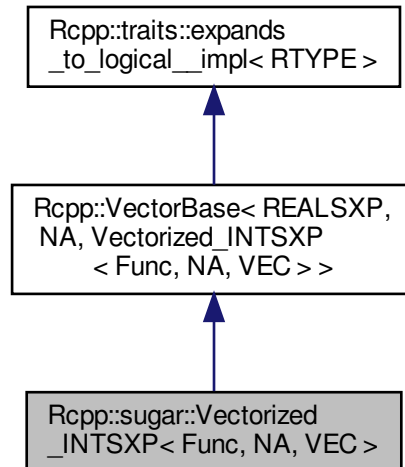
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/Vectorized\_Math.h

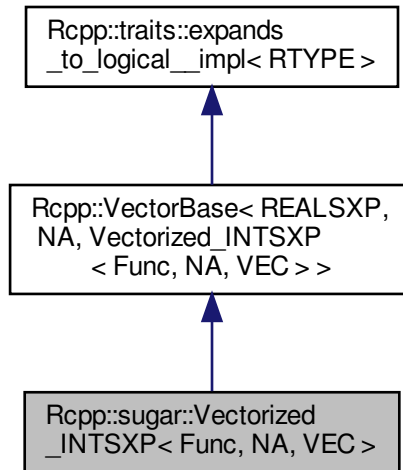
## 6.816 Rcpp::sugar::Vectorized\_INTSXP< Func, NA, VEC > Class Template Reference

```
#include <Vectorized_Math.h>
```

Inheritance diagram for Rcpp::sugar::Vectorized\_INTSXP< Func, NA, VEC >:



Collaboration diagram for `Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >`:



## Public Types

- typedef `Rcpp::VectorBase< INTSXP, NA, VEC >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< INTSXP, NA, VEC >::type` `VEC_EXT`

## Public Member Functions

- `Vectorized_INTSXP` (const `VEC_TYPE` &object\_)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const `VEC_EXT` & `object`

### 6.816.1 Detailed Description

```

template<DDFun Func, bool NA, typename VEC>
class Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >
  
```

Definition at line 47 of file `Vectorized_Math.h`.

## 6.816.2 Member Typedef Documentation

### 6.816.2.1 VEC\_EXT

```
template<DDFun Func, bool NA, typename VEC >
typedef Rcpp::traits::Extractor<INTSXP,NA,VEC>::type Rcpp::sugar::Vectorized_INTSXP< Func, NA,
VEC >::VEC_EXT
```

Definition at line 50 of file Vectorized\_Math.h.

### 6.816.2.2 VEC\_TYPE

```
template<DDFun Func, bool NA, typename VEC >
typedef Rcpp::VectorBase<INTSXP,NA,VEC> Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >::VEC_TYPE
```

Definition at line 49 of file Vectorized\_Math.h.

## 6.816.3 Constructor & Destructor Documentation

### 6.816.3.1 Vectorized\_INTSXP()

```
template<DDFun Func, bool NA, typename VEC >
Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >::Vectorized_INTSXP (
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 52 of file Vectorized\_Math.h.

## 6.816.4 Member Function Documentation

### 6.816.4.1 operator[]()

```
template<DDFun Func, bool NA, typename VEC >
double Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 53 of file Vectorized\_Math.h.

### 6.816.4.2 size()

```
template<DDFun Func, bool NA, typename VEC >
R_xlen_t Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >::size ( ) const [inline]
```

Definition at line 58 of file Vectorized\_Math.h.

## 6.816.5 Member Data Documentation

### 6.816.5.1 object

```
template<DDFun Func, bool NA, typename VEC >
const VEC_EXT& Rcpp::sugar::Vectorized_INTSXP< Func, NA, VEC >::object [private]
```

Definition at line 61 of file Vectorized\_Math.h.

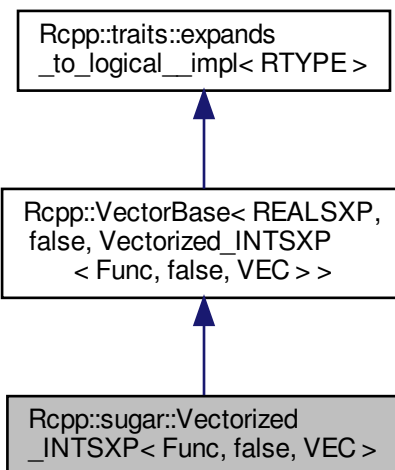
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/block/Vectorized\_Math.h

## 6.817 Rcpp::sugar::Vectorized\_INTSXP< Func, false, VEC > Class Template Reference

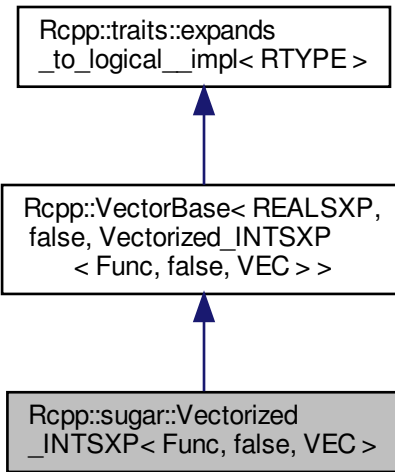
```
#include <Vectorized_Math.h>
```

Inheritance diagram for Rcpp::sugar::Vectorized\_INTSXP< Func, false, VEC >:





Collaboration diagram for Rcpp::sugar::Vectorized\_INTSXP< Func, false, VEC >:



## Public Types

- typedef `Rcpp::VectorBase< INTSXP, false, VEC >` `VEC_TYPE`
- typedef `Rcpp::traits::Extractor< INTSXP, false, VEC >::type` `VEC_EXT`

## Public Member Functions

- `Vectorized_INTSXP` (const `VEC_TYPE` &object\_)
- double `operator[]` (R\_xlen\_t i) const
- R\_xlen\_t `size` () const

## Private Attributes

- const `VEC_EXT` & `object`

### 6.817.1 Detailed Description

```

template<DDFun Func, typename VEC>
class Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >
  
```

Definition at line 64 of file `Vectorized_Math.h`.

## 6.817.2 Member Typedef Documentation

### 6.817.2.1 VEC\_EXT

```
template<DDFun Func, typename VEC >
typedef Rcpp::traits::Extractor<INTSXP,false,VEC>::type Rcpp::sugar::Vectorized_INTSXP< Func,
false, VEC >::VEC_EXT
```

Definition at line 68 of file Vectorized\_Math.h.

### 6.817.2.2 VEC\_TYPE

```
template<DDFun Func, typename VEC >
typedef Rcpp::VectorBase<INTSXP,false,VEC> Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >↔
::VEC_TYPE
```

Definition at line 67 of file Vectorized\_Math.h.

## 6.817.3 Constructor & Destructor Documentation

### 6.817.3.1 Vectorized\_INTSXP()

```
template<DDFun Func, typename VEC >
Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >::Vectorized_INTSXP (
    const VEC_TYPE & object_ ) [inline]
```

Definition at line 70 of file Vectorized\_Math.h.

## 6.817.4 Member Function Documentation

### 6.817.4.1 operator[]()

```
template<DDFun Func, typename VEC >
double Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >::operator[] (
    R_xlen_t i ) const [inline]
```

Definition at line 71 of file Vectorized\_Math.h.

### 6.817.4.2 size()

```
template<DDFun Func, typename VEC >  
R_xlen_t Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >::size ( ) const [inline]
```

Definition at line 74 of file Vectorized\_Math.h.

## 6.817.5 Member Data Documentation

### 6.817.5.1 object

```
template<DDFun Func, typename VEC >  
const VEC_EXT& Rcpp::sugar::Vectorized_INTSXP< Func, false, VEC >::object [private]
```

Definition at line 77 of file Vectorized\_Math.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/sugar/block/Vectorized\\_Math.h](#)

## 6.818 Rcpp::void\_type Struct Reference

```
#include <get_return_type.h>
```

### 6.818.1 Detailed Description

Definition at line 27 of file get\_return\_type.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/module/get\\_return\\_type.h](#)

## 6.819 Rcpp::traits::void\_wrap\_tag Struct Reference

```
#include <module_wrap_traits.h>
```

### 6.819.1 Detailed Description

Definition at line 30 of file `module_wrap_traits.h`.

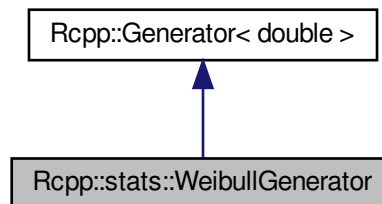
The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/module_wrap_traits.h`

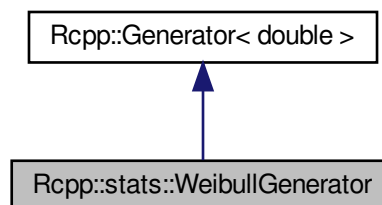
## 6.820 Rcpp::stats::WeibullGenerator Class Reference

```
#include <rweibull.h>
```

Inheritance diagram for `Rcpp::stats::WeibullGenerator`:



Collaboration diagram for `Rcpp::stats::WeibullGenerator`:



### Public Member Functions

- `WeibullGenerator` (`double shape_`, `double scale_`)
- `double operator()` () const

## Private Attributes

- double [shape\\_inv](#)
- double [scale](#)

## Additional Inherited Members

### 6.820.1 Detailed Description

Definition at line 28 of file rweibull.h.

### 6.820.2 Constructor & Destructor Documentation

#### 6.820.2.1 WeibullGenerator()

```
Rcpp::stats::WeibullGenerator::WeibullGenerator (  
    double shape_,  
    double scale_ ) [inline]
```

Definition at line 31 of file rweibull.h.

### 6.820.3 Member Function Documentation

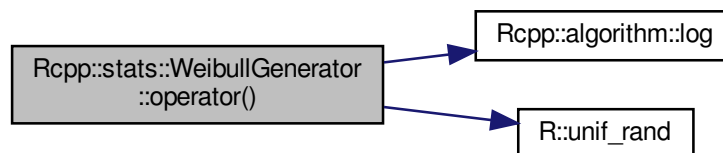
#### 6.820.3.1 operator>()

```
double Rcpp::stats::WeibullGenerator::operator() ( ) const [inline]
```

Definition at line 34 of file rweibull.h.

References [Rcpp::algorithm::log\(\)](#), [scale](#), [shape\\_inv](#), and [R::unif\\_rand\(\)](#).

Here is the call graph for this function:



## 6.820.4 Member Data Documentation

### 6.820.4.1 scale

```
double Rcpp::stats::WeibullGenerator::scale [private]
```

Definition at line 39 of file rweibull.h.

Referenced by operator(()).

### 6.820.4.2 shape\_inv

```
double Rcpp::stats::WeibullGenerator::shape_inv [private]
```

Definition at line 39 of file rweibull.h.

Referenced by operator(()).

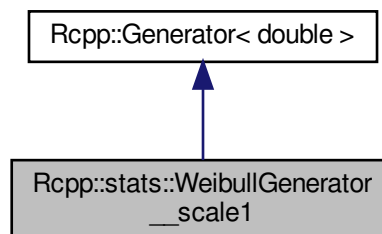
The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rweibull.h](#)

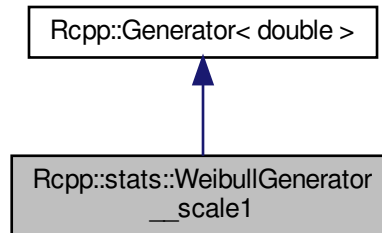
## 6.821 Rcpp::stats::WeibullGenerator\_\_scale1 Class Reference

```
#include <rweibull.h>
```

Inheritance diagram for Rcpp::stats::WeibullGenerator\_\_scale1:



Collaboration diagram for Rcpp::stats::WeibullGenerator\_\_scale1:



## Public Member Functions

- [WeibullGenerator\\_\\_scale1](#) (double shape\_)
- double [operator\(\)](#) () const

## Private Attributes

- double [shape\\_inv](#)

## Additional Inherited Members

### 6.821.1 Detailed Description

Definition at line 42 of file `rweibull.h`.

### 6.821.2 Constructor & Destructor Documentation

#### 6.821.2.1 WeibullGenerator\_\_scale1()

```
Rcpp::stats::WeibullGenerator__scale1::WeibullGenerator__scale1 (  
    double shape_ ) [inline]
```

Definition at line 45 of file `rweibull.h`.

### 6.821.3 Member Function Documentation

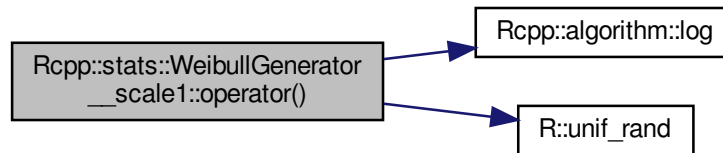
#### 6.821.3.1 operator()

```
double Rcpp::stats::WeibullGenerator__scale1::operator() ( ) const [inline]
```

Definition at line 48 of file rweibull.h.

References `Rcpp::algorithm::log()`, `shape_inv`, and `R::unif_rand()`.

Here is the call graph for this function:



### 6.821.4 Member Data Documentation

#### 6.821.4.1 shape\_inv

```
double Rcpp::stats::WeibullGenerator__scale1::shape_inv [private]
```

Definition at line 53 of file rweibull.h.

Referenced by `operator()`.

The documentation for this class was generated from the following file:

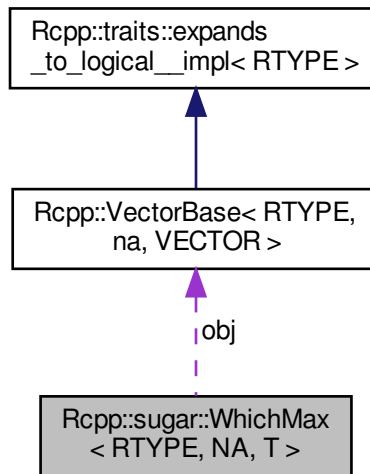
- [inst/include/Rcpp/stats/random/rweibull.h](#)



## 6.822 Rcpp::sugar::WhichMax< RTYPE, NA, T > Class Template Reference

```
#include <which_max.h>
```

Collaboration diagram for Rcpp::sugar::WhichMax< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`

### Public Member Functions

- `WhichMax` (const `VEC_TYPE` &obj\_)
- `R_xlen_t get` () const

### Private Attributes

- const `VEC_TYPE` &obj

### 6.822.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::WhichMax< RTYPE, NA, T >
```

Definition at line 29 of file `which_max.h`.

## 6.822.2 Member Typedef Documentation

### 6.822.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::WhichMax< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file which\_max.h.

### 6.822.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::WhichMax< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file which\_max.h.

## 6.822.3 Constructor & Destructor Documentation

### 6.822.3.1 WhichMax()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::WhichMax< RTYPE, NA, T >::WhichMax (
    const VEC_TYPE & obj_ ) [inline]
```

Definition at line 33 of file which\_max.h.

## 6.822.4 Member Function Documentation

### 6.822.4.1 get()

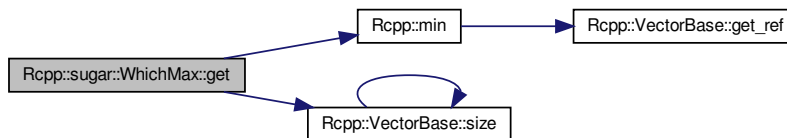
```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::WhichMax< RTYPE, NA, T >::get ( ) const [inline]
```

Definition at line 35 of file which\_max.h.

References Rcpp::min(), Rcpp::sugar::WhichMax< RTYPE, NA, T >::obj, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Referenced by Rcpp::which\_max().

Here is the call graph for this function:



## 6.822.5 Member Data Documentation

### 6.822.5.1 obj

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::WhichMax< RTYPE, NA, T >::obj [private]
```

Definition at line 53 of file which\_max.h.

Referenced by Rcpp::sugar::WhichMax< RTYPE, NA, T >::get(), and Rcpp::sugar::WhichMax< RTYPE, false, T >::get().

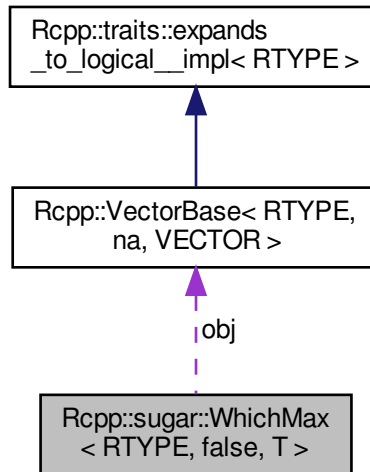
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/which\_max.h

## 6.823 Rcpp::sugar::WhichMax< RTYPE, false, T > Class Template Reference

```
#include <which_max.h>
```

Collaboration diagram for Rcpp::sugar::WhichMax< RTYPE, false, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, false, T >` `VEC_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`

### Public Member Functions

- `WhichMax` (const `VEC_TYPE` &obj\_)
- `R_xlen_t get` () const

### Private Attributes

- const `VEC_TYPE` &obj

## 6.823.1 Detailed Description

```
template<int RTYPE, typename T>
class Rcpp::sugar::WhichMax< RTYPE, false, T >
```

Definition at line 58 of file which\_max.h.

## 6.823.2 Member Typedef Documentation

### 6.823.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::WhichMax< RTYPE, false, T >::STORAGE
```

Definition at line 61 of file which\_max.h.

### 6.823.2.2 VEC\_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::WhichMax< RTYPE, false, T >::VEC_TYPE
```

Definition at line 60 of file which\_max.h.

## 6.823.3 Constructor & Destructor Documentation

### 6.823.3.1 WhichMax()

```
template<int RTYPE, typename T >
Rcpp::sugar::WhichMax< RTYPE, false, T >::WhichMax (
    const VEC_TYPE & obj_ ) [inline]
```

Definition at line 62 of file which\_max.h.

## 6.823.4 Member Function Documentation

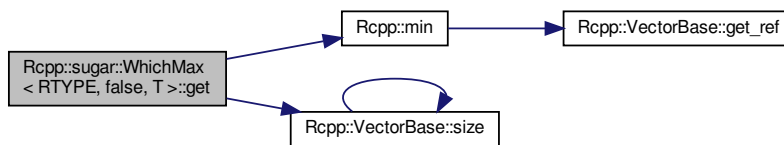
### 6.823.4.1 get()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::WhichMax< RTYPE, false, T >::get ( ) const [inline]
```

Definition at line 64 of file which\_max.h.

References `Rcpp::min()`, `Rcpp::sugar::WhichMax< RTYPE, NA, T >::obj`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`.

Here is the call graph for this function:



## 6.823.5 Member Data Documentation

### 6.823.5.1 obj

```
template<int RTYPE, typename T >
const VEC_TYPE& Rcpp::sugar::WhichMax< RTYPE, false, T >::obj [private]
```

Definition at line 80 of file which\_max.h.

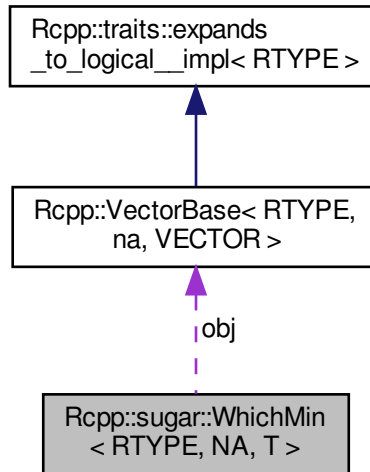
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/which_max.h`

## 6.824 Rcpp::sugar::WhichMin< RTYPE, NA, T > Class Template Reference

```
#include <which_min.h>
```

Collaboration diagram for Rcpp::sugar::WhichMin< RTYPE, NA, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, NA, T >` `VEC_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`

### Public Member Functions

- `WhichMin` (const `VEC_TYPE` &obj\_)
- `R_xlen_t get` () const

### Private Attributes

- const `VEC_TYPE` &obj

### 6.824.1 Detailed Description

```
template<int RTYPE, bool NA, typename T>
class Rcpp::sugar::WhichMin< RTYPE, NA, T >
```

Definition at line 29 of file `which_min.h`.

## 6.824.2 Member Typedef Documentation

### 6.824.2.1 STORAGE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::WhichMin< RTYPE, NA, T >::STORAGE
```

Definition at line 32 of file which\_min.h.

### 6.824.2.2 VEC\_TYPE

```
template<int RTYPE, bool NA, typename T >
typedef Rcpp::VectorBase<RTYPE,NA,T> Rcpp::sugar::WhichMin< RTYPE, NA, T >::VEC_TYPE
```

Definition at line 31 of file which\_min.h.

## 6.824.3 Constructor & Destructor Documentation

### 6.824.3.1 WhichMin()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::WhichMin< RTYPE, NA, T >::WhichMin (
    const VEC_TYPE & obj_ ) [inline]
```

Definition at line 33 of file which\_min.h.

## 6.824.4 Member Function Documentation



### 6.824.4.1 get()

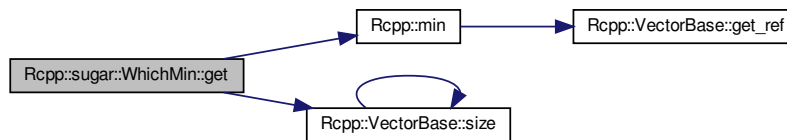
```
template<int RTYPE, bool NA, typename T >
R_xlen_t Rcpp::sugar::WhichMin< RTYPE, NA, T >::get ( ) const [inline]
```

Definition at line 35 of file which\_min.h.

References Rcpp::min(), Rcpp::sugar::WhichMin< RTYPE, NA, T >::obj, and Rcpp::VectorBase< RTYPE, na, VECTOR >::size().

Referenced by Rcpp::which\_min().

Here is the call graph for this function:



## 6.824.5 Member Data Documentation

### 6.824.5.1 obj

```
template<int RTYPE, bool NA, typename T >
const VEC_TYPE& Rcpp::sugar::WhichMin< RTYPE, NA, T >::obj [private]
```

Definition at line 53 of file which\_min.h.

Referenced by Rcpp::sugar::WhichMin< RTYPE, NA, T >::get(), and Rcpp::sugar::WhichMin< RTYPE, false, T >::get().

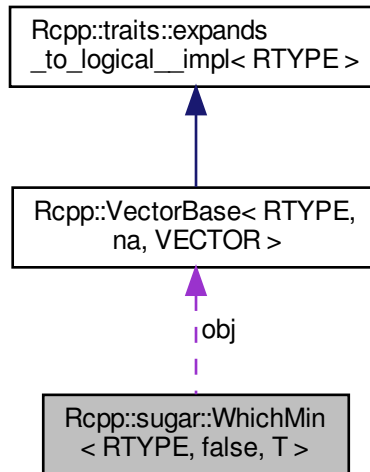
The documentation for this class was generated from the following file:

- inst/include/Rcpp/sugar/functions/which\_min.h

## 6.825 Rcpp::sugar::WhichMin< RTYPE, false, T > Class Template Reference

```
#include <which_min.h>
```

Collaboration diagram for Rcpp::sugar::WhichMin< RTYPE, false, T >:



### Public Types

- typedef `Rcpp::VectorBase< RTYPE, false, T >` `VEC_TYPE`
- typedef `Rcpp::traits::storage_type< RTYPE >::type` `STORAGE`

### Public Member Functions

- `WhichMin` (const `VEC_TYPE` &obj\_)
- `R_xlen_t get` () const

### Private Attributes

- const `VEC_TYPE` &obj

## 6.825.1 Detailed Description

```
template<int RTYPE, typename T>
class Rcpp::sugar::WhichMin< RTYPE, false, T >
```

Definition at line 58 of file which\_min.h.

## 6.825.2 Member Typedef Documentation

### 6.825.2.1 STORAGE

```
template<int RTYPE, typename T >
typedef Rcpp::traits::storage_type<RTYPE>::type Rcpp::sugar::WhichMin< RTYPE, false, T >::STORAGE
```

Definition at line 61 of file which\_min.h.

### 6.825.2.2 VEC\_TYPE

```
template<int RTYPE, typename T >
typedef Rcpp::VectorBase<RTYPE,false,T> Rcpp::sugar::WhichMin< RTYPE, false, T >::VEC_TYPE
```

Definition at line 60 of file which\_min.h.

## 6.825.3 Constructor & Destructor Documentation

### 6.825.3.1 WhichMin()

```
template<int RTYPE, typename T >
Rcpp::sugar::WhichMin< RTYPE, false, T >::WhichMin (
    const VEC_TYPE & obj_ ) [inline]
```

Definition at line 62 of file which\_min.h.

## 6.825.4 Member Function Documentation

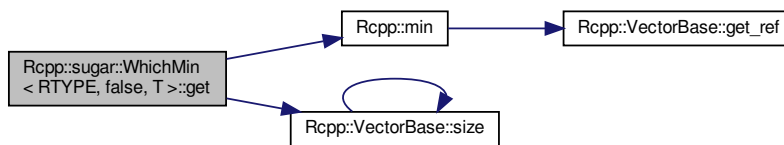
### 6.825.4.1 get()

```
template<int RTYPE, typename T >
R_xlen_t Rcpp::sugar::WhichMin< RTYPE, false, T >::get ( ) const [inline]
```

Definition at line 64 of file which\_min.h.

References `Rcpp::min()`, `Rcpp::sugar::WhichMin< RTYPE, NA, T >::obj`, and `Rcpp::VectorBase< RTYPE, na, VECTOR >::size()`.

Here is the call graph for this function:



## 6.825.5 Member Data Documentation

### 6.825.5.1 obj

```
template<int RTYPE, typename T >
const VEC_TYPE& Rcpp::sugar::WhichMin< RTYPE, false, T >::obj [private]
```

Definition at line 80 of file which\_min.h.

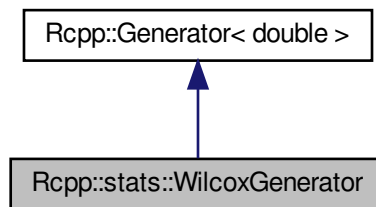
The documentation for this class was generated from the following file:

- `inst/include/Rcpp/sugar/functions/which_min.h`

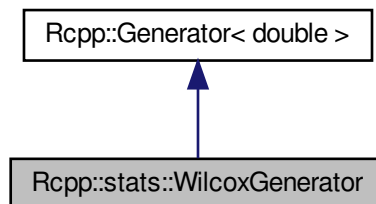
## 6.826 Rcpp::stats::WilcoxGenerator Class Reference

```
#include <rwilcox.h>
```

Inheritance diagram for Rcpp::stats::WilcoxGenerator:



Collaboration diagram for Rcpp::stats::WilcoxGenerator:



### Public Member Functions

- [WilcoxGenerator](#) (double mm\_, double nn\_)
- double [operator\(\)](#) () const

### Private Attributes

- double [mm](#)
- double [nn](#)

## Additional Inherited Members

### 6.826.1 Detailed Description

Definition at line 28 of file `rwilcox.h`.

### 6.826.2 Constructor & Destructor Documentation

#### 6.826.2.1 WilcoxGenerator()

```
Rcpp::stats::WilcoxGenerator::WilcoxGenerator (
    double mm_,
    double nn_ ) [inline]
```

Definition at line 30 of file `rwilcox.h`.

### 6.826.3 Member Function Documentation

#### 6.826.3.1 operator>()()

```
double Rcpp::stats::WilcoxGenerator::operator() ( ) const [inline]
```

Definition at line 31 of file `rwilcox.h`.

References `mm`, and `nn`.

### 6.826.4 Member Data Documentation

#### 6.826.4.1 mm

```
double Rcpp::stats::WilcoxGenerator::mm [private]
```

Definition at line 33 of file `rwilcox.h`.

Referenced by `operator>()()`.

### 6.826.4.2 nn

```
double Rcpp::stats::WilcoxGenerator::nn [private]
```

Definition at line 33 of file `rwilcox.h`.

Referenced by `operator()`.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/stats/random/rwilcox.h](#)

## 6.827 Rcpp::traits::wrap\_type\_char\_array Struct Reference

```
#include <wrap_type_traits.h>
```

### 6.827.1 Detailed Description

character arrays

Definition at line 57 of file `wrap_type_traits.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/wrap\\_type\\_traits.h](#)

## 6.828 Rcpp::traits::wrap\_type\_enum\_tag Struct Reference

```
#include <wrap_type_traits.h>
```

### 6.828.1 Detailed Description

enums

Definition at line 52 of file `wrap_type_traits.h`.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/wrap\\_type\\_traits.h](#)

## 6.829 Rcpp::traits::wrap\_type\_module\_object\_pointer\_tag Struct Reference

```
#include <wrap_type_traits.h>
```

### 6.829.1 Detailed Description

module objects pointers (object<T>). conversion done using [make\\_new\\_object<>\(\)](#)

Definition at line 42 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[wrap\\_type\\_traits.h](#)

## 6.830 Rcpp::traits::wrap\_type\_module\_object\_tag Struct Reference

```
#include <wrap_type_traits.h>
```

### 6.830.1 Detailed Description

declared module object type (by the RCPP\_EXPOSED\_CLASS macro)

Definition at line 47 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[wrap\\_type\\_traits.h](#)

## 6.831 Rcpp::traits::wrap\_type\_primitive\_tag Struct Reference

```
#include <wrap_type_traits.h>
```

### 6.831.1 Detailed Description

primitive type : int, double, std::string, Rcomplex, size\_t, Rbyte

Definition at line 32 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[wrap\\_type\\_traits.h](#)



## 6.832 Rcpp::traits::wrap\_type\_traits< T > Struct Template Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_unknown\\_tag](#) [wrap\\_category](#)

### 6.832.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::wrap_type_traits< T >
```

Type trait that helps the dispatch of wrap to the proper method

This builds a struct that contains a typedef called `wrap_category` that has to be one of "wrap\_type\_primitive\_tag" or "wrap\_type\_unknown\_tag"

The default is "wrap\_type\_unknown\_tag" and this is specialized for primitive types

Definition at line 68 of file `wrap_type_traits.h`.

### 6.832.2 Member Typedef Documentation

#### 6.832.2.1 wrap\_category

```
template<typename T >  
typedef wrap\_type\_unknown\_tag Rcpp::traits::wrap\_type\_traits< T >::wrap\_category
```

Definition at line 68 of file `wrap_type_traits.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/wrap_type_traits.h`

## 6.833 Rcpp::traits::wrap\_type\_traits< bool > Struct Reference

```
#include <wrap_type_traits.h>
```

## Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) [wrap\\_category](#)

### 6.833.1 Detailed Description

Definition at line 82 of file `wrap_type_traits.h`.

### 6.833.2 Member Typedef Documentation

#### 6.833.2.1 `wrap_category`

```
typedef wrap\_type\_primitive\_tag Rcpp::traits::wrap\_type\_traits< bool >::wrap\_category
```

Definition at line 82 of file `wrap_type_traits.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/wrap_type_traits.h`

## 6.834 `Rcpp::traits::wrap_type_traits< char >` Struct Reference

```
#include <wrap_type_traits.h>
```

## Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) [wrap\\_category](#)

### 6.834.1 Detailed Description

Definition at line 86 of file `wrap_type_traits.h`.

### 6.834.2 Member Typedef Documentation

### 6.834.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< char >::wrap_category
```

Definition at line 86 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.835 Rcpp::traits::wrap\_type\_traits< char[N]> Struct Template Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_char\\_array](#) wrap\_category

### 6.835.1 Detailed Description

```
template<std::size_t N>  
struct Rcpp::traits::wrap_type_traits< char[N]>
```

Definition at line 70 of file wrap\_type\_traits.h.

### 6.835.2 Member Typedef Documentation

#### 6.835.2.1 wrap\_category

```
template<std::size_t N>  
typedef wrap_type_char_array Rcpp::traits::wrap_type_traits< char[N]>::wrap_category
```

Definition at line 70 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.836 Rcpp::traits::wrap\_type\_traits< const char[N]> Struct Template Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_char\\_array](#) [wrap\\_category](#)

### 6.836.1 Detailed Description

```
template<std::size_t N>  
struct Rcpp::traits::wrap_type_traits< const char[N]>
```

Definition at line 71 of file `wrap_type_traits.h`.

### 6.836.2 Member Typedef Documentation

#### 6.836.2.1 wrap\_category

```
template<std::size_t N>  
typedef wrap\_type\_char\_array Rcpp::traits::wrap\_type\_traits< const char\[N\]>::wrap\_category
```

Definition at line 71 of file `wrap_type_traits.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/wrap_type_traits.h`

## 6.837 Rcpp::traits::wrap\_type\_traits< const int > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) [wrap\\_category](#)

### 6.837.1 Detailed Description

Definition at line 76 of file wrap\_type\_traits.h.

### 6.837.2 Member Typedef Documentation

#### 6.837.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< const int >::wrap_category
```

Definition at line 76 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.838 Rcpp::traits::wrap\_type\_traits< double > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef wrap\_type\_primitive\_tag wrap\_category

### 6.838.1 Detailed Description

Definition at line 78 of file wrap\_type\_traits.h.

### 6.838.2 Member Typedef Documentation

#### 6.838.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< double >::wrap_category
```

Definition at line 78 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.839 Rcpp::traits::wrap\_type\_traits< float > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) wrap\_category

### 6.839.1 Detailed Description

Definition at line 89 of file wrap\_type\_traits.h.

### 6.839.2 Member Typedef Documentation

#### 6.839.2.1 wrap\_category

```
typedef wrap\_type\_primitive\_tag Rcpp::traits::wrap_type_traits< float >::wrap_category
```

Definition at line 89 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/[wrap\\_type\\_traits.h](#)

## 6.840 Rcpp::traits::wrap\_type\_traits< int > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) wrap\_category

### 6.840.1 Detailed Description

Total specialization for primitive types

Definition at line 75 of file wrap\_type\_traits.h.

## 6.840.2 Member Typedef Documentation

### 6.840.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< int >::wrap_category
```

Definition at line 75 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.841 Rcpp::traits::wrap\_type\_traits< long > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef wrap\_type\_primitive\_tag wrap\_category

### 6.841.1 Detailed Description

Definition at line 93 of file wrap\_type\_traits.h.

## 6.841.2 Member Typedef Documentation

### 6.841.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< long >::wrap_category
```

Definition at line 93 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.842 Rcpp::traits::wrap\_type\_traits< long double > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) [wrap\\_category](#)

### 6.842.1 Detailed Description

Definition at line 96 of file `wrap_type_traits.h`.

### 6.842.2 Member Typedef Documentation

#### 6.842.2.1 `wrap_category`

```
typedef wrap\_type\_primitive\_tag Rcpp::traits::wrap\_type\_traits< long double >::wrap\_category
```

Definition at line 96 of file `wrap_type_traits.h`.

The documentation for this struct was generated from the following file:

- `inst/include/Rcpp/traits/wrap_type_traits.h`

## 6.843 Rcpp::traits::wrap\_type\_traits< Rbyte > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) [wrap\\_category](#)

### 6.843.1 Detailed Description

Definition at line 79 of file `wrap_type_traits.h`.



## 6.843.2 Member Typedef Documentation

### 6.843.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< Rbyte >::wrap_category
```

Definition at line 79 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.844 Rcpp::traits::wrap\_type\_traits< Rcomplex > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef wrap\_type\_primitive\_tag wrap\_category

### 6.844.1 Detailed Description

Definition at line 80 of file wrap\_type\_traits.h.

## 6.844.2 Member Typedef Documentation

### 6.844.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< Rcomplex >::wrap_category
```

Definition at line 80 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.845 Rcpp::traits::wrap\_type\_traits< Rcpp::Date > Struct Reference

```
#include <Date.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) [wrap\\_category](#)

### 6.845.1 Detailed Description

Definition at line 28 of file Date.h.

### 6.845.2 Member Typedef Documentation

#### 6.845.2.1 wrap\_category

```
typedef wrap\_type\_primitive\_tag Rcpp::traits::wrap\_type\_traits< Rcpp::Date >::wrap_category
```

Definition at line 29 of file Date.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/api/bones/Date.h](#)

## 6.846 Rcpp::traits::wrap\_type\_traits< Rcpp::Datetime > Struct Reference

```
#include <Datetime.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) [wrap\\_category](#)

### 6.846.1 Detailed Description

Definition at line 28 of file Datetime.h.

## 6.846.2 Member Typedef Documentation

### 6.846.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< Rcpp::Datetime >::wrap_category
```

Definition at line 29 of file Datetime.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/api/bones/Datetime.h

## 6.847 Rcpp::traits::wrap\_type\_traits< Rcpp::object< T > > Struct Template Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef wrap\_type\_module\_object\_pointer\_tag wrap\_category

### 6.847.1 Detailed Description

```
template<typename T>  
struct Rcpp::traits::wrap_type_traits< Rcpp::object< T > >
```

Definition at line 101 of file wrap\_type\_traits.h.

## 6.847.2 Member Typedef Documentation

### 6.847.2.1 wrap\_category

```
template<typename T >  
typedef wrap_type_module_object_pointer_tag Rcpp::traits::wrap_type_traits< Rcpp::object< T > >  
>::wrap_category
```

Definition at line 101 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.848 Rcpp::traits::wrap\_type\_traits< Rcpp::String > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) [wrap\\_category](#)

### 6.848.1 Detailed Description

Definition at line 85 of file [wrap\\_type\\_traits.h](#).

### 6.848.2 Member Typedef Documentation

#### 6.848.2.1 wrap\_category

```
typedef wrap\_type\_primitive\_tag Rcpp::traits::wrap\_type\_traits< Rcpp::String >::wrap\_category
```

Definition at line 85 of file [wrap\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/wrap\\_type\\_traits.h](#)

## 6.849 Rcpp::traits::wrap\_type\_traits< short > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) [wrap\\_category](#)

### 6.849.1 Detailed Description

Definition at line 98 of file [wrap\\_type\\_traits.h](#).

## 6.849.2 Member Typedef Documentation

### 6.849.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< short >::wrap_category
```

Definition at line 98 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/wrap\\_type\\_traits.h](#)

## 6.850 Rcpp::traits::wrap\_type\_traits< std::complex< double > > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) wrap\_category

### 6.850.1 Detailed Description

Definition at line 91 of file wrap\_type\_traits.h.

## 6.850.2 Member Typedef Documentation

### 6.850.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< std::complex< double > >::wrap_category
```

Definition at line 91 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/wrap\\_type\\_traits.h](#)

## 6.851 Rcpp::traits::wrap\_type\_traits< std::complex< float > > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) wrap\_category

#### 6.851.1 Detailed Description

Definition at line 90 of file wrap\_type\_traits.h.

#### 6.851.2 Member Typedef Documentation

##### 6.851.2.1 wrap\_category

```
typedef wrap\_type\_primitive\_tag Rcpp::traits::wrap_type_traits< std::complex< float > >::wrap_category
```

Definition at line 90 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.852 Rcpp::traits::wrap\_type\_traits< std::string > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) wrap\_category

#### 6.852.1 Detailed Description

Definition at line 83 of file wrap\_type\_traits.h.

## 6.852.2 Member Typedef Documentation

### 6.852.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< std::string >::wrap_category
```

Definition at line 83 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.853 Rcpp::traits::wrap\_type\_traits< std::wstring > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef wrap\_type\_primitive\_tag wrap\_category

### 6.853.1 Detailed Description

Definition at line 84 of file wrap\_type\_traits.h.

## 6.853.2 Member Typedef Documentation

### 6.853.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< std::wstring >::wrap_category
```

Definition at line 84 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.854 Rcpp::traits::wrap\_type\_traits< unsigned int > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) [wrap\\_category](#)

#### 6.854.1 Detailed Description

Definition at line 81 of file [wrap\\_type\\_traits.h](#).

#### 6.854.2 Member Typedef Documentation

##### 6.854.2.1 wrap\_category

```
typedef wrap\_type\_primitive\_tag Rcpp::traits::wrap\_type\_traits< unsigned int >::wrap_category
```

Definition at line 81 of file [wrap\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/wrap\\_type\\_traits.h](#)

## 6.855 Rcpp::traits::wrap\_type\_traits< unsigned long > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) [wrap\\_category](#)

#### 6.855.1 Detailed Description

Definition at line 94 of file [wrap\\_type\\_traits.h](#).



## 6.855.2 Member Typedef Documentation

### 6.855.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< unsigned long >::wrap_category
```

Definition at line 94 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.856 Rcpp::traits::wrap\_type\_traits< unsigned short > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef wrap\_type\_primitive\_tag wrap\_category

### 6.856.1 Detailed Description

Definition at line 99 of file wrap\_type\_traits.h.

## 6.856.2 Member Typedef Documentation

### 6.856.2.1 wrap\_category

```
typedef wrap_type_primitive_tag Rcpp::traits::wrap_type_traits< unsigned short >::wrap_category
```

Definition at line 99 of file wrap\_type\_traits.h.

The documentation for this struct was generated from the following file:

- inst/include/Rcpp/traits/wrap\_type\_traits.h

## 6.857 Rcpp::traits::wrap\_type\_traits< wchar\_t > Struct Reference

```
#include <wrap_type_traits.h>
```

### Public Types

- typedef [wrap\\_type\\_primitive\\_tag](#) [wrap\\_category](#)

### 6.857.1 Detailed Description

Definition at line 87 of file [wrap\\_type\\_traits.h](#).

### 6.857.2 Member Typedef Documentation

#### 6.857.2.1 wrap\_category

```
typedef wrap\_type\_primitive\_tag Rcpp::traits::wrap\_type\_traits< wchar_t >::wrap_category
```

Definition at line 87 of file [wrap\\_type\\_traits.h](#).

The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/wrap\\_type\\_traits.h](#)

## 6.858 Rcpp::traits::wrap\_type\_unknown\_tag Struct Reference

```
#include <wrap_type_traits.h>
```

### 6.858.1 Detailed Description

unknown. Implicit conversion to SEXP will be tried.

Definition at line 37 of file [wrap\\_type\\_traits.h](#).

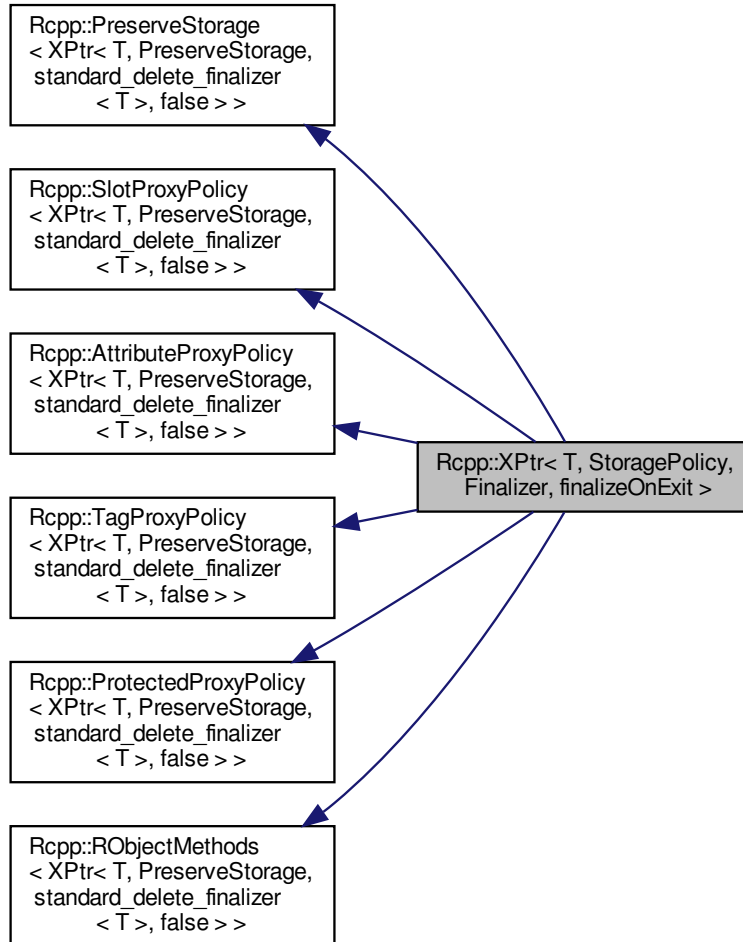
The documentation for this struct was generated from the following file:

- [inst/include/Rcpp/traits/wrap\\_type\\_traits.h](#)

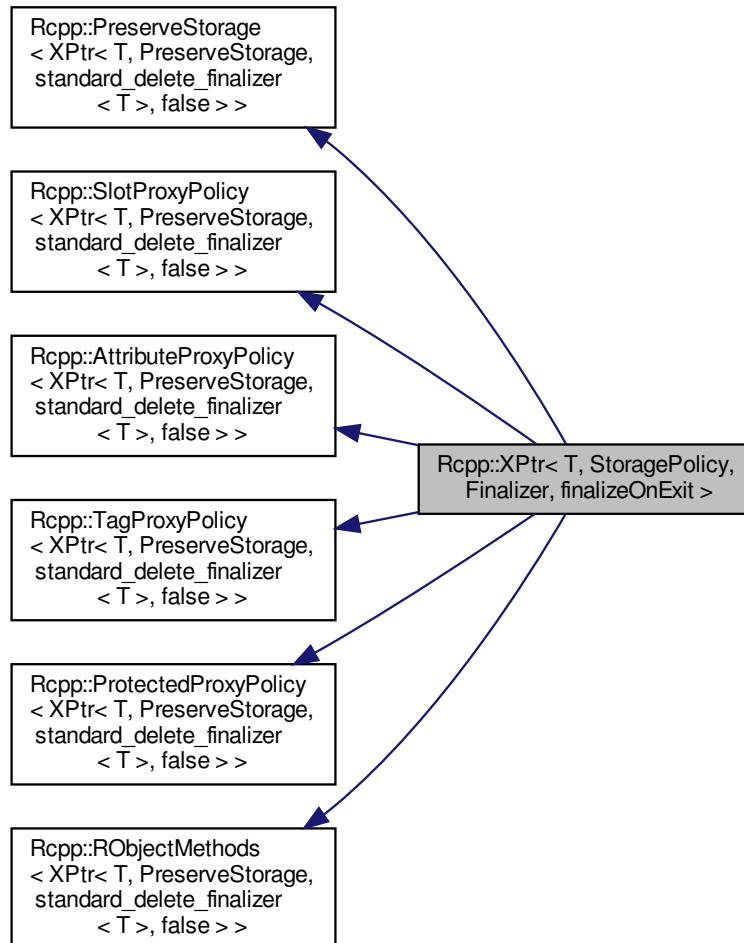
## 6.859 Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > Class Template Reference

```
#include <XPtr.h>
```

Inheritance diagram for Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >:



Collaboration diagram for `Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >`:



## Public Types

- `typedef StoragePolicy< XPtr > Storage`
- `typedef void(* unspecified_bool_type) ()`

## Public Member Functions

- `XPtr` (SEXP x)
- `XPtr` (SEXP x, SEXP tag, SEXP prot)
- `XPtr` (T \*p, bool set\_delete\_finalizer=true, SEXP tag=R\_NilValue, SEXP prot=R\_NilValue)
- `XPtr` (const XPtr &other)
- `XPtr & operator=` (const XPtr &other)

- T \* [get](#) () const
- [operator unspecified\\_bool\\_type](#) () const
- bool [operator!](#) () const
- T \* [checked\\_get](#) () const
- T & [operator\\*](#) () const
- T \* [operator->](#) () const
- void [setDeleteFinalizer](#) ()
- void [release](#) ()
- [operator T\\*](#) ()
- void [update](#) (SEXP)

## Static Public Member Functions

- static void [unspecified\\_bool\\_true](#) ()

## Private Member Functions

- void [checked\\_set](#) (SEXP x)

### 6.859.1 Detailed Description

```
template<typename T, template< class > class StoragePolicy = PreserveStorage, void Finalizer = standard_delete_finalizer<T>,
bool finalizeOnExit = false>
class Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >
```

Definition at line 61 of file XPtr.h.

### 6.859.2 Member Typedef Documentation

#### 6.859.2.1 Storage

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
typedef StoragePolicy<XPtr> Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::Storage
```

Definition at line 71 of file XPtr.h.

### 6.859.2.2 unspecified\_bool\_type

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
typedef void(* Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::unspecified_bool_type)
()
```

Boolean operator wrapper for `get()` using the "safe bool idiom", see: [http://www.boost.org/doc/libs/1\\_57\\_0/boost/smart\\_ptr/detail/operator\\_bool.hpp](http://www.boost.org/doc/libs/1_57_0/boost/smart_ptr/detail/operator_bool.hpp)

Definition at line 134 of file XPtr.h.

## 6.859.3 Constructor & Destructor Documentation

### 6.859.3.1 XPtr() [1/4]

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::XPtr (
    SEXP x ) [inline], [explicit]
```

constructs a `XPtr` wrapping the external pointer (EXTPTRSXP SEXP)

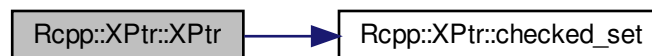
#### Parameters

<i>xp</i>	external pointer to wrap
-----------	--------------------------

Definition at line 78 of file XPtr.h.

References `Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked_set()`.

Here is the call graph for this function:



## 6.859.3.2 XPtr() [2/4]

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::XPtr (
    SEXP x,
    SEXP tag,
    SEXP prot ) [inline], [explicit]
```

constructs a [XPtr](#) wrapping the external pointer (EXTPTRSXP SEXP)

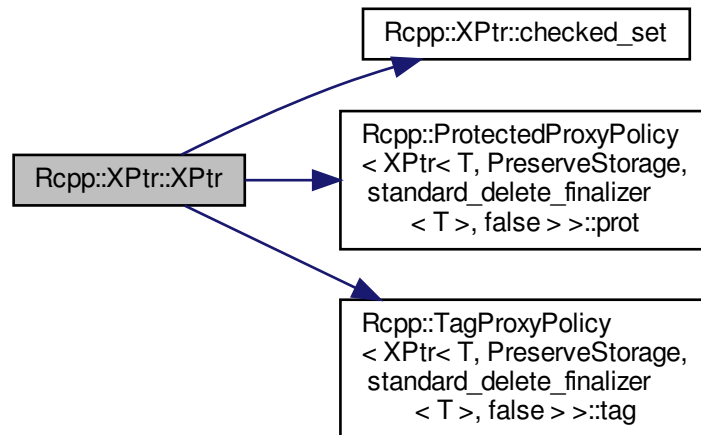
## Parameters

<i>xp</i>	external pointer to wrap
<i>tag</i>	tag to assign to external pointer
<i>prot</i>	protected data to assign to external pointer

Definition at line 87 of file XPtr.h.

References [Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked\\_set\(\)](#), [Rcpp::ProtectedProxyPolicy< XPtr< T, PreserveStorage, standard\\_delete\\_finalizer< T >, false > >::prot\(\)](#), and [Rcpp::TagProxyPolicy< XPtr< T, PreserveStorage, standard\\_delete\\_finalizer< T >, false > >::tag\(\)](#).

Here is the call graph for this function:



### 6.859.3.3 XPtr() [3/4]

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::XPtr (
    T * p,
    bool set_delete_finalizer = true,
    SEXP tag = R_NilValue,
    SEXP prot = R_NilValue ) [inline], [explicit]
```

creates a new external pointer wrapping the dumb pointer p.

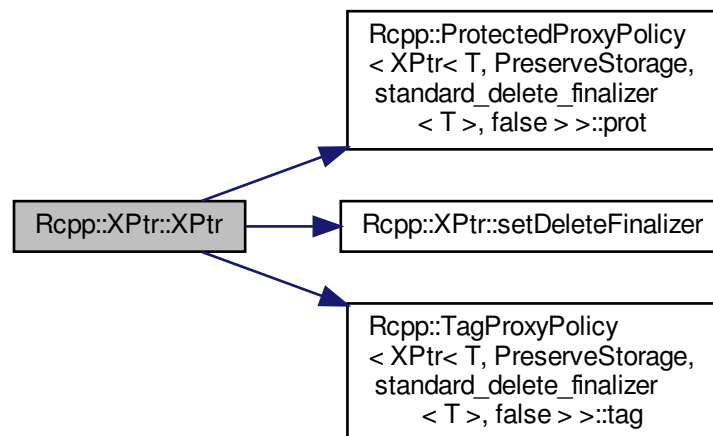
#### Parameters

<i>p</i>	dumb pointer to some object
<i>set_delete_finalizer</i>	if set to true, a finalizer will be registered for the external pointer. The finalizer is called when the xp is garbage collected. The finalizer is merely a call to the delete operator or the pointer so you need to make sure the pointer can be "delete" d this way (has to be a C++ object)

Definition at line 104 of file XPtr.h.

References Rcpp::ProtectedProxyPolicy< XPtr< T, PreserveStorage, standard\_delete\_finalizer< T >, false > >::prot(), RCPP\_DEBUG\_2, Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::setDeleteFinalizer(), and Rcpp::TagProxyPolicy< XPtr< T, PreserveStorage, standard\_delete\_finalizer< T >, false > >::tag().

Here is the call graph for this function:





### 6.859.3.4 XPtr() [4/4]

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::XPtr (
    const XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > & other ) [inline]
```

Definition at line 113 of file XPtr.h.

## 6.859.4 Member Function Documentation

### 6.859.4.1 checked\_get()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
T* Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked_get ( ) const [inline]
```

Access underlying pointer throwing an exception if the ptr is NULL

Definition at line 146 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::get().

Referenced by Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator T\*(), Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator\*(), and Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator->().

Here is the call graph for this function:



### 6.859.4.2 checked\_set()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
void Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked_set (
    SEXP x ) [inline], [private]
```

Definition at line 202 of file XPtr.h.

Referenced by Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::XPtr().

### 6.859.4.3 get()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
T* Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::get ( ) const [inline]
```

Typesafe accessor for underlying pointer (use `checked_get` if you want an exception thrown if the pointer is NULL)

Definition at line 126 of file XPtr.h.

Referenced by `Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked_get()`, `Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator unspecified_bool_type()`, `Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator!()`, and `Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::release()`.

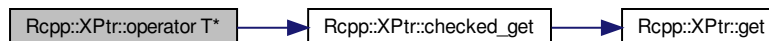
### 6.859.4.4 operator T\*()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator T* ( ) [inline]
```

Definition at line 195 of file XPtr.h.

References `Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked_get()`.

Here is the call graph for this function:



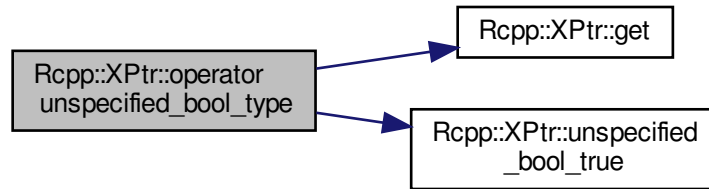
### 6.859.4.5 operator unspecified\_bool\_type()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator unspecified_bool_type ( )
const [inline]
```

Definition at line 136 of file XPtr.h.

References `Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::get()`, and `Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::unspecified_bool_true()`.

Here is the call graph for this function:



#### 6.859.4.6 operator"!()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =  
standard_delete_finalizer<T>, bool finalizeOnExit = false>  
bool Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator! ( ) const [inline]
```

Definition at line 139 of file XPtr.h.

References `Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::get()`.

Here is the call graph for this function:



### 6.859.4.7 operator\*()

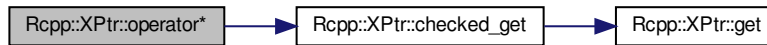
```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
T& Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator* ( ) const [inline]
```

Returns a reference to the object wrapped. This allows this object to look and feel like a dumb pointer to T

Definition at line 157 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked\_get().

Here is the call graph for this function:



### 6.859.4.8 operator->()

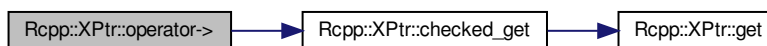
```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
T* Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator-> ( ) const [inline]
```

Returns the dumb pointer. This allows to call the `->` operator on this as if it was the dumb pointer

Definition at line 165 of file XPtr.h.

References Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::checked\_get().

Here is the call graph for this function:



#### 6.859.4.9 operator=()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
XPtr& Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator= (
    const XPtr< T, StoragePolicy, Finalizer, finalizeOnExit > & other ) [inline]
```

Definition at line 117 of file XPtr.h.

#### 6.859.4.10 release()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
void Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::release ( ) [inline]
```

Release the external pointer (if any) immediately. This will cause the pointer to be deleted and its storage to be set to NULL. After this call the `get()` method returns NULL and the `checked_get()` method throws an exception.

See the discussion here for the basic logic behind release: <https://stat.ethz.ch/pipermail/r-help/2007-August/137871.html>

Definition at line 183 of file XPtr.h.

References `Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::get()`.

Here is the call graph for this function:



#### 6.859.4.11 setDeleteFinalizer()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
void Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::setDeleteFinalizer ( ) [inline]
```

Definition at line 169 of file XPtr.h.

Referenced by `Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::XPtr()`.

### 6.859.4.12 unspecified\_bool\_true()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
static void Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::unspecified_bool_true ( )
[inline], [static]
```

Definition at line 135 of file XPtr.h.

Referenced by Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::operator unspecified\_bool\_type().

### 6.859.4.13 update()

```
template<typename T , template< class > class StoragePolicy = PreserveStorage, void Finalizer =
standard_delete_finalizer<T>, bool finalizeOnExit = false>
void Rcpp::XPtr< T, StoragePolicy, Finalizer, finalizeOnExit >::update (
    SEXP ) [inline]
```

Definition at line 199 of file XPtr.h.

The documentation for this class was generated from the following file:

- [inst/include/Rcpp/XPtr.h](#)

## 6.860 Rcpp::traits::zero\_type< T > Class Template Reference

```
#include <one_type.h>
```

### Public Member Functions

- [operator T \(\)](#) const

### Private Member Functions

- Rcomplex [op \(true\\_type\)](#) const
- T [op \(false\\_type\)](#) const

### 6.860.1 Detailed Description

```
template<typename T>
class Rcpp::traits::zero_type< T >
```

Definition at line 55 of file one\_type.h.

## 6.860.2 Member Function Documentation

### 6.860.2.1 op() [1/2]

```
template<typename T >
T Rcpp::traits::zero_type< T >::op (
    false_type ) const [inline], [private]
```

Definition at line 64 of file one\_type.h.

### 6.860.2.2 op() [2/2]

```
template<typename T >
Rcomplex Rcpp::traits::zero_type< T >::op (
    true_type ) const [inline], [private]
```

Definition at line 57 of file one\_type.h.

Referenced by Rcpp::traits::zero\_type< T >::operator T().

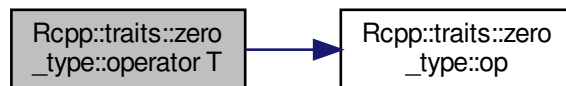
### 6.860.2.3 operator T()

```
template<typename T >
Rcpp::traits::zero_type< T >::operator T ( ) const [inline]
```

Definition at line 69 of file one\_type.h.

References Rcpp::traits::zero\_type< T >::op().

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- inst/include/Rcpp/traits/[one\\_type.h](#)





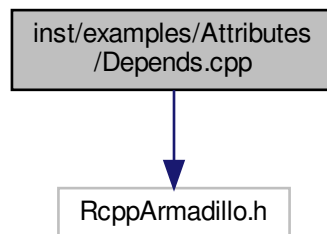
## Chapter 7

# File Documentation

### 7.1 inst/examples/Attributes/cppFunction.R File Reference

### 7.2 inst/examples/Attributes/Depends.cpp File Reference

```
#include <RcppArmadillo.h>  
Include dependency graph for Depends.cpp:
```



### Functions

- [List fastLm](#) ([NumericVector](#) yr, [NumericMatrix](#) Xr)

#### 7.2.1 Function Documentation

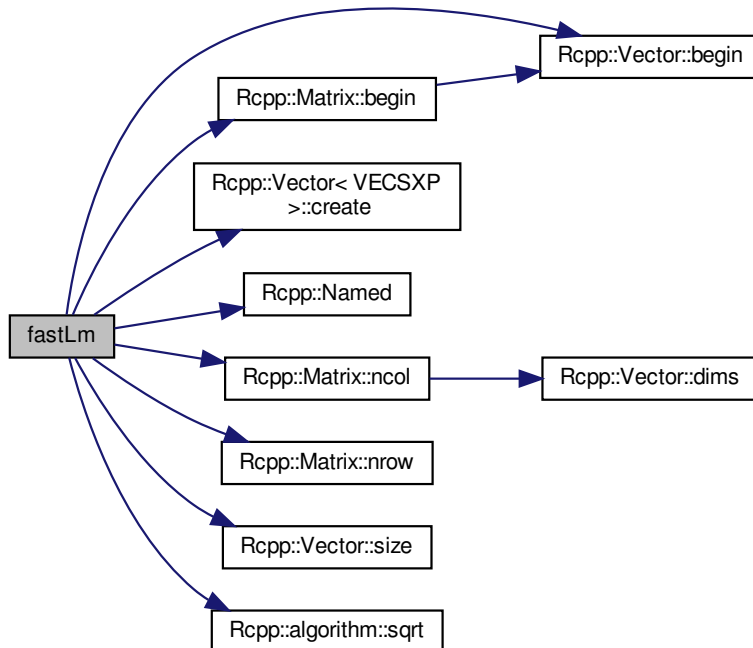
### 7.2.1.1 fastLm()

```
List fastLm (
    NumericVector yr,
    NumericMatrix Xr )
```

Definition at line 9 of file Depends.cpp.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Matrix< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< VECSXP >::create()`, `Rcpp::Named()`, `Rcpp::Matrix< RTYPE, StoragePolicy >::ncol()`, `Rcpp::Matrix< RTYPE, StoragePolicy >::nrow()`, `Rcpp::Vector< RTYPE, StoragePolicy >::size()`, and `Rcpp::algorithm::sqrt()`.

Here is the call graph for this function:



## 7.3 inst/examples/Attributes/Export.cpp File Reference

```
#include <Rcpp.h>
```

Include dependency graph for `Export.cpp`:



## Functions

- int `fibonacci` (const int x)
- `NumericVector` `convolve` (`NumericVector` a, `NumericVector` b)
- `List` `lapplyCpp` (`List` input, `Function` f)

### 7.3.1 Function Documentation

#### 7.3.1.1 `convolve()`

```
NumericVector convolve (  
    NumericVector a,  
    NumericVector b )
```

Definition at line 17 of file `Export.cpp`.

References `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



#### 7.3.1.2 `fibonacci()`

```
int fibonacci (  
    const int x )
```

Definition at line 7 of file `Export.cpp`.

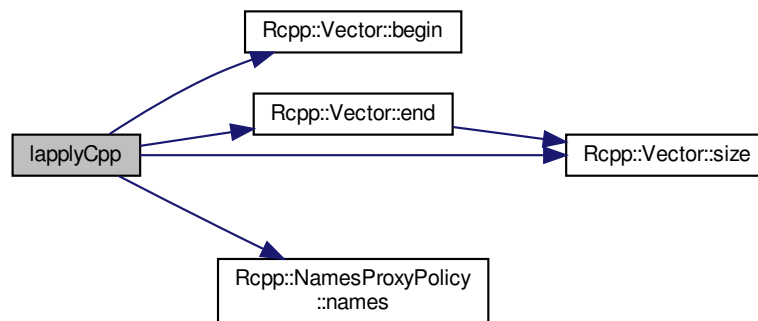
### 7.3.1.3 lapplyCpp()

```
List lapplyCpp (
    List input,
    Function f )
```

Definition at line 32 of file Export.cpp.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::NamesProxyPolicy< CLASS >::names()`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



## 7.4 inst/examples/Attributes/sourceCpp.R File Reference

## 7.5 inst/examples/ConvolveBenchmarks/convolve10\_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "convolve10_cpp.h"
#include "loopmacro.h"
```

Include dependency graph for `convolve10_cpp.cpp`:



## Functions

- `RcppExport SEXP convolve10cpp (SEXP a, SEXP b)`

## 7.5.1 Function Documentation

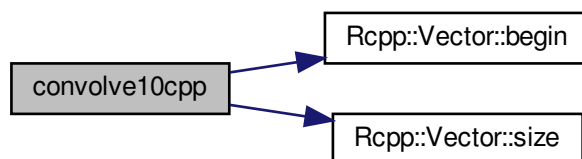
### 7.5.1.1 convolve10cpp()

```
RcppExport SEXP convolve10cpp (  
    SEXP a,  
    SEXP b )
```

Definition at line 17 of file convolve10\_cpp.cpp.

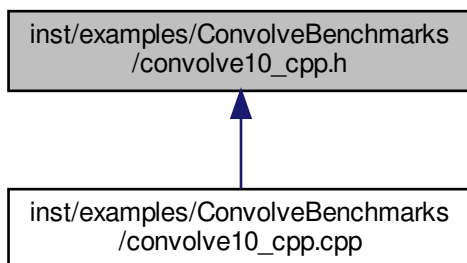
References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



## 7.6 inst/examples/ConvolveBenchmarks/convolve10\_cpp.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Cache](#)
- class [Vec](#)

## 7.7 inst/examples/ConvolveBenchmarks/convolve11\_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
Include dependency graph for convolve11_cpp.cpp:
```



## Functions

- [RcppExport](#) SEXP [convolve11cpp](#) (SEXP a, SEXP b)

### 7.7.1 Function Documentation

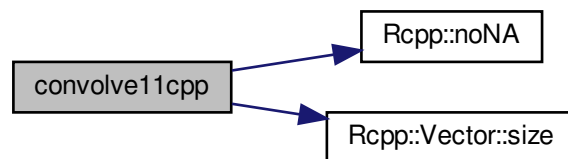
#### 7.7.1.1 convolve11cpp()

```
RcppExport SEXP convolve11cpp (
    SEXP a,
    SEXP b )
```

Definition at line 10 of file convolve11\_cpp.cpp.

References [Rcpp::noNA\(\)](#), and [Rcpp::Vector< RTYPE, StoragePolicy >::size\(\)](#).

Here is the call graph for this function:



## 7.8 inst/examples/ConvolveBenchmarks/convolve12\_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
Include dependency graph for convolve12_cpp.cpp:
```



### Functions

- [RcppExport](#) SEXP [convolve12cpp](#) (SEXP a, SEXP b)

### 7.8.1 Function Documentation

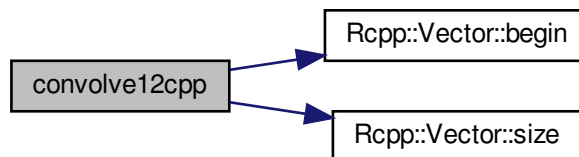
#### 7.8.1.1 convolve12cpp()

```
RcppExport SEXP convolve12cpp (
    SEXP a,
    SEXP b )
```

Definition at line 7 of file convolve12\_cpp.cpp.

References [Rcpp::Vector< RTYPE, StoragePolicy >::begin\(\)](#), and [Rcpp::Vector< RTYPE, StoragePolicy >::size\(\)](#).

Here is the call graph for this function:



## 7.9 inst/examples/ConvolveBenchmarks/convolve13\_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
Include dependency graph for convolve13_cpp.cpp:
```



### Functions

- `template<typename T >`  
`T convolve (const T &a, const T &b)`
- `RcppExport` `SEXP convolve13cpp (SEXP a, SEXP b)`

### 7.9.1 Function Documentation

#### 7.9.1.1 convolve()

```
template<typename T >
T convolve (
    const T & a,
    const T & b )
```

Definition at line 8 of file convolve13\_cpp.cpp.

Referenced by convolve13cpp().

#### 7.9.1.2 convolve13cpp()

```
RcppExport SEXP convolve13cpp (
    SEXP a,
    SEXP b )
```

Definition at line 21 of file convolve13\_cpp.cpp.

References convolve().



Here is the call graph for this function:



## 7.10 inst/examples/ConvolveBenchmarks/convolve14\_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
Include dependency graph for convolve14_cpp.cpp:
```



### Functions

- [RcppExport](#) [SEXP](#) [convolve14cpp](#) ([SEXP](#) a, [SEXP](#) b)

### 7.10.1 Function Documentation

#### 7.10.1.1 convolve14cpp()

```
RcppExport SEXP convolve14cpp (
    SEXP a,
    SEXP b )
```

Definition at line 8 of file convolve14\_cpp.cpp.

References [Rcpp::Vector< RTYPE, StoragePolicy >::size\(\)](#).

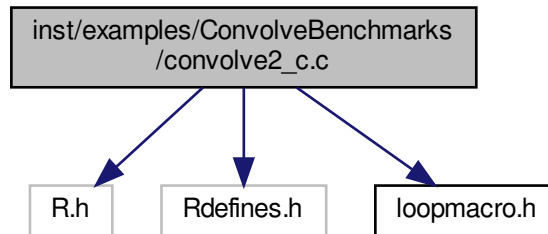
Here is the call graph for this function:



## 7.11 inst/examples/ConvolveBenchmarks/convolve2\_c.c File Reference

```
#include <R.h>  
#include <Rdefines.h>  
#include "loopmacro.h"
```

Include dependency graph for convolve2\_c.c:



### Functions

- SEXP [convolve2](#) (SEXP a, SEXP b)

#### 7.11.1 Function Documentation

### 7.11.1.1 convolve2()

```
SEXP convolve2 (  
    SEXP a,  
    SEXP b )
```

#### Examples

[ConvolveBenchmarks/convolve2\\_c.c](#).

Definition at line 7 of file convolve2\_c.c.

## 7.12 inst/examples/ConvolveBenchmarks/convolve3\_cpp.cpp File Reference

```
#include <Rcpp.h>  
#include "loopmacro.h"  
Include dependency graph for convolve3_cpp.cpp:
```



### Functions

- [RcppExport](#) SEXP [convolve3cpp](#) (SEXP a, SEXP b)

### 7.12.1 Function Documentation

#### 7.12.1.1 convolve3cpp()

```
RcppExport SEXP convolve3cpp (  
    SEXP a,  
    SEXP b )
```

#### Examples

[ConvolveBenchmarks/convolve3\\_cpp.cpp](#).

Definition at line 7 of file convolve3\_cpp.cpp.

References `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



## 7.13 inst/examples/ConvolveBenchmarks/convolve4\_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
Include dependency graph for convolve4_cpp.cpp:
```



### Functions

- `RcppExport` `SEXP convolve4cpp` (`SEXP a`, `SEXP b`)

### 7.13.1 Function Documentation

#### 7.13.1.1 convolve4cpp()

```
RcppExport SEXP convolve4cpp (
    SEXP a,
    SEXP b )
```

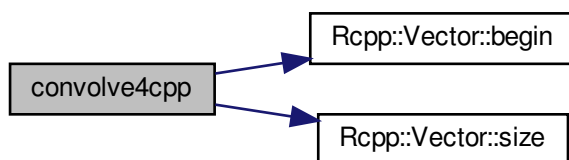
### Examples

[ConvolveBenchmarks/convolve4\\_cpp.cpp](#).

Definition at line 7 of file convolve4\_cpp.cpp.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



## 7.14 inst/examples/ConvolveBenchmarks/convolve5\_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
Include dependency graph for convolve5_cpp.cpp:
```



### Functions

- [RcppExport](#) `SEXP convolve5cpp` (`SEXP a`, `SEXP b`)

#### 7.14.1 Function Documentation

### 7.14.1.1 convolve5cpp()

```
RcppExport SEXP convolve5cpp (
    SEXP a,
    SEXP b )
```

Definition at line 9 of file convolve5\_cpp.cpp.

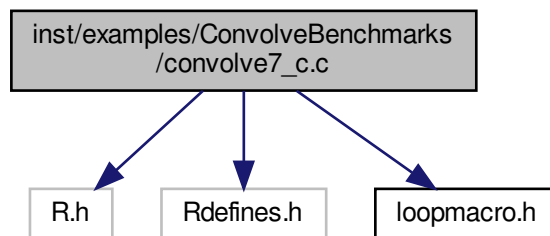
References Rcpp::Vector< RTYPE, StoragePolicy >::size().

Here is the call graph for this function:



## 7.15 inst/examples/ConvolveBenchmarks/convolve7\_c.c File Reference

```
#include <R.h>
#include <Rdefines.h>
#include "loopmacro.h"
Include dependency graph for convolve7_c.c:
```



### Functions

- SEXP `convolve7` (SEXP a, SEXP b)

## 7.15.1 Function Documentation

### 7.15.1.1 convolve7()

```
SEXP convolve7 (  
    SEXP a,  
    SEXP b )
```

#### Examples

[ConvolveBenchmarks/convolve7\\_c.c](#).

Definition at line 8 of file convolve7\_c.c.

## 7.16 inst/examples/ConvolveBenchmarks/convolve8\_cpp.cpp File Reference

```
#include <Rcpp.h>  
#include "loopmacro.h"  
Include dependency graph for convolve8_cpp.cpp:
```



### Classes

- class [Vec](#)

### Functions

- [RcppExport](#) SEXP [convolve8cpp](#) (SEXP a, SEXP b)

### 7.16.1 Function Documentation

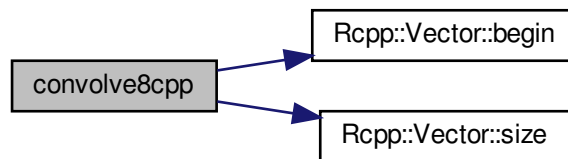
### 7.16.1.1 convolve8cpp()

```
RcppExport SEXP convolve8cpp (
    SEXP a,
    SEXP b )
```

Definition at line 24 of file convolve8\_cpp.cpp.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



## 7.17 inst/examples/ConvolveBenchmarks/convolve9\_cpp.cpp File Reference

```
#include <Rcpp.h>
#include "loopmacro.h"
Include dependency graph for convolve9_cpp.cpp:
```



### Classes

- class [Cache](#)
- class [Vec](#)

### Functions

- `RcppExport` `SEXP convolve9cpp` (`SEXP a`, `SEXP b`)



## 7.17.1 Function Documentation

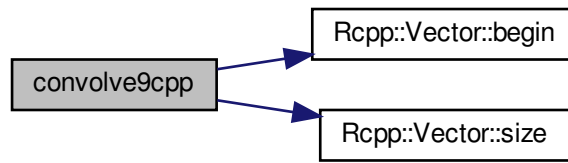
### 7.17.1.1 convolve9cpp()

```
RcppExport SEXP convolve9cpp (
    SEXP a,
    SEXP b )
```

Definition at line 42 of file convolve9\_cpp.cpp.

References `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, and `Rcpp::Vector< RTYPE, StoragePolicy >::size()`.

Here is the call graph for this function:



## 7.18 inst/examples/ConvolveBenchmarks/exampleRCode.r File Reference

## 7.19 inst/examples/ConvolveBenchmarks/loopmacro.h File Reference

This graph shows which files directly or indirectly include this file:



## Macros

- `#define LOOPMACRO_C(name)`
- `#define LOOPMACRO_CPP(name) RcppExport LOOPMACRO_C(name)`

## 7.19.1 Macro Definition Documentation

### 7.19.1.1 LOOPMACRO\_C

```
#define LOOPMACRO_C(
    name )
```

#### Value:

```
SEXP name##_loop(SEXP n_, SEXP a, SEXP b){ \
    int n = INTEGER(n_)[0] ;           \
    SEXP res = R_NilValue ;           \
    for( int i=0; i<n; i++){           \
        res = name( a, b ) ;           \
    }                                   \
    return res ;                       \
}
```

#### Examples

[ConvolveBenchmarks/convolve2\\_c.c](#), and [ConvolveBenchmarks/convolve7\\_c.c](#).

Definition at line 2 of file loopmacro.h.

### 7.19.1.2 LOOPMACRO\_CPP

```
#define LOOPMACRO_CPP(
    name ) RcppExport LOOPMACRO_C(name)
```

#### Examples

[ConvolveBenchmarks/convolve3\\_cpp.cpp](#), and [ConvolveBenchmarks/convolve4\\_cpp.cpp](#).

Definition at line 12 of file loopmacro.h.

## 7.20 inst/examples/ConvolveBenchmarks/overhead.r File Reference

## 7.21 inst/examples/ConvolveBenchmarks/overhead\_1.cpp File Reference

```
#include <Rcpp.h>
```

Include dependency graph for overhead\_1.cpp:



## Functions

- SEXP [overhead\\_cpp](#) (SEXP a, SEXP b)
- void [R\\_init\\_overhead\\_1](#) (DllInfo \*info)

### 7.21.1 Function Documentation

#### 7.21.1.1 overhead\_cpp()

```
SEXP overhead_cpp (  
    SEXP a,  
    SEXP b )
```

Definition at line 8 of file overhead\_1.cpp.

Referenced by [R\\_init\\_overhead\\_1\(\)](#).

#### 7.21.1.2 R\_init\_overhead\_1()

```
void R_init_overhead_1 (  
    DllInfo * info )
```

Definition at line 12 of file overhead\_1.cpp.

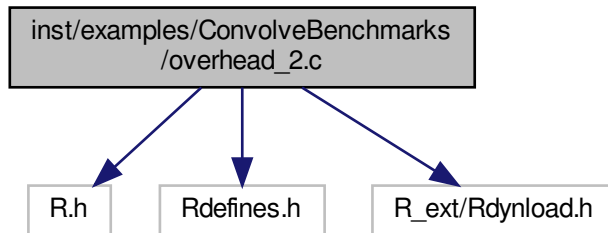
References [overhead\\_cpp\(\)](#).

Here is the call graph for this function:



## 7.22 inst/examples/ConvolveBenchmarks/overhead\_2.c File Reference

```
#include <R.h>
#include <Rdefines.h>
#include <R_ext/Rdynload.h>
Include dependency graph for overhead_2.c:
```



### Functions

- SEXP `overhead_c` (SEXP `a`, SEXP `b`)
- void `R_init_overhead_2` (DllInfo `*info`)

### 7.22.1 Function Documentation

#### 7.22.1.1 `overhead_c()`

```
SEXP overhead_c (
    SEXP a,
    SEXP b )
```

Definition at line 8 of file `overhead_2.c`.

Referenced by `R_init_overhead_2()`.

### 7.22.1.2 R\_init\_overhead\_2()

```
void R_init_overhead_2 (  
    DllInfo * info )
```

Definition at line 12 of file overhead\_2.c.

References overhead\_c().

Here is the call graph for this function:



## 7.23 inst/examples/FastLM/benchmark.r File Reference

## 7.24 inst/examples/FastLM/benchmarkLongley.r File Reference

## 7.25 inst/examples/FastLM/fastLMviaArmadillo.r File Reference

## 7.26 inst/examples/FastLM/fastLMviaGSL.r File Reference

## 7.27 inst/examples/FastLM/lmArmadillo.R File Reference

## 7.28 inst/examples/FastLM/lmGSL.R File Reference

## 7.29 inst/examples/functionCallback/newApiExample.r File Reference

## 7.30 inst/examples/Misc/fibonacci.r File Reference

## 7.31 inst/examples/Misc/ifelseLooped.r File Reference

## 7.32 inst/examples/Misc/newFib.r File Reference

## 7.33 inst/examples/Misc/piBySimulation.r File Reference

## 7.34 inst/examples/Misc/piSugar.cpp File Reference

```
#include <Rcpp.h>
```

Include dependency graph for piSugar.cpp:



## Functions

- double [piSugar](#) (const int N)

### 7.34.1 Function Documentation

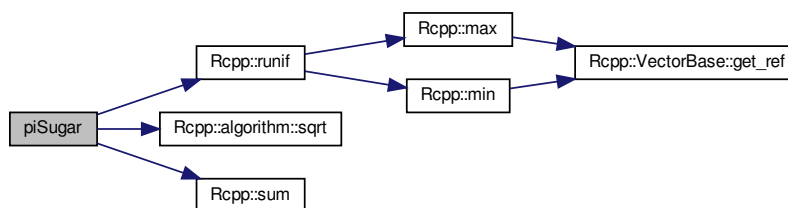
#### 7.34.1.1 piSugar()

```
double piSugar (
    const int N )
```

Definition at line 7 of file piSugar.cpp.

References [Rcpp::runif\(\)](#), [Rcpp::algorithm::sqrt\(\)](#), and [Rcpp::sum\(\)](#).

Here is the call graph for this function:



## 7.35 inst/examples/OpenMP/check.R File Reference

## 7.36 inst/examples/OpenMP/OpenMPandInline.r File Reference

## 7.37 inst/examples/OpenMP/piWithInterrupts.cpp File Reference

```
#include <Rcpp.h>
#include <R_ext/Utils.h>
Include dependency graph for piWithInterrupts.cpp:
```



### Classes

- class [interrupt\\_exception](#)

### Functions

- static void [check\\_interrupt\\_impl](#) (void \*)
- bool [check\\_interrupt](#) ()
- [RcppExport](#) SEXP [PiLeibniz](#) (SEXP n, SEXP frequency)

### 7.37.1 Function Documentation

#### 7.37.1.1 [check\\_interrupt\(\)](#)

```
bool check_interrupt ( ) [inline]
```

Call this method to check for user interrupts. This is based on the results of a discussion on the R-devel mailing list, suggested by Simon Urbanek.

#### Attention

This method must not be called by any other thread than the master thread. If called from within an OpenMP parallel for loop, make sure to check for `omp_get_thread_num()==0` before calling this method!

**Returns**

True, if a user interrupt has been detected.

Definition at line 64 of file piWithInterrupts.cpp.

References `check_interrupt_impl()`.

Referenced by `PiLeibniz()`.

Here is the call graph for this function:

**7.37.1.2 check\_interrupt\_impl()**

```
static void check_interrupt_impl (  
    void * ) [inline], [static]
```

Do the actual check for an interrupt.

**Attention**

This method should never be called directly.

**Parameters**

in	<i>dummy</i>	Dummy argument.
----	--------------	-----------------

Definition at line 50 of file piWithInterrupts.cpp.

Referenced by `check_interrupt()`.



### 7.37.1.3 PiLeibniz()

```
RcppExport SEXP PiLeibniz (  
    SEXP n,  
    SEXP frequency )
```

Compute pi using the Leibniz formula (a very inefficient approach).

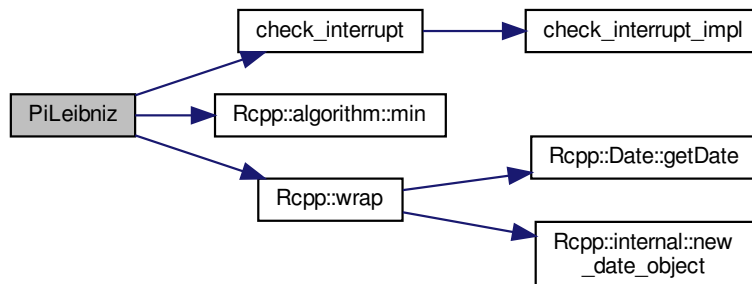
#### Parameters

in	<i>n</i>	Number of summands
in	<i>frequency</i>	Check for interrupts after every <i>frequency</i> loop cycles.

Definition at line 75 of file piWithInterrupts.cpp.

References BEGIN\_RCPP, check\_interrupt(), END\_RCPP, Rcpp::algorithm::min(), and Rcpp::wrap().

Here is the call graph for this function:

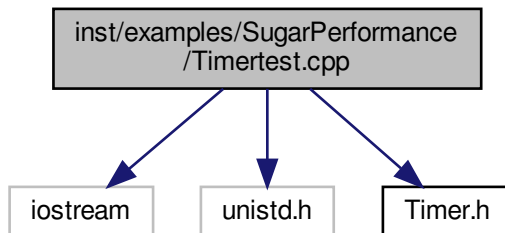


- 7.38** [inst/examples/performance/extractors.R](#) File Reference
- 7.39** [inst/examples/performance/performance.R](#) File Reference
- 7.40** [inst/examples/RcppGibbs/RcppGibbs.R](#) File Reference
- 7.41** [inst/examples/RcppGibbs/RcppGibbs\\_Updated.R](#) File Reference
- 7.42** [inst/examples/RcppGibbs/timeRNGs.R](#) File Reference
- 7.43** [inst/examples/RcppInline/external\\_pointer.r](#) File Reference
- 7.44** [inst/examples/RcppInline/RcppInlineExample.r](#) File Reference
- 7.45** [inst/examples/RcppInline/RcppInlineWithLibsExamples.r](#) File Reference
- 7.46** [inst/examples/RcppInline/RcppSimpleExample.r](#) File Reference
- 7.47** [inst/examples/RcppInline/RObject.r](#) File Reference
- 7.48** [inst/examples/RcppInline/UncaughtExceptions.r](#) File Reference
- 7.49** [inst/examples/SugarPerformance/sugarBenchmarks.R](#) File Reference
- 7.50** [inst/examples/SugarPerformance/Timertest.cpp](#) File Reference

```
#include <iostream>  
#include <unistd.h>
```

```
#include "Timer.h"
```

Include dependency graph for Timertest.cpp:



## Functions

- int [main](#) ()

### 7.50.1 Function Documentation

#### 7.50.1.1 main()

```
int main ( )
```

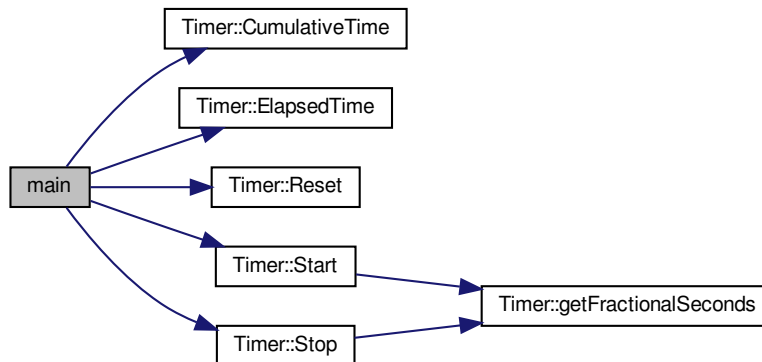
#### Examples

[SugarPerformance/Timertest.cpp](#).

Definition at line 9 of file `Timertest.cpp`.

References `Timer::CumulativeTime()`, `Timer::ElapsedTime()`, `Timer::Reset()`, `Timer::Start()`, and `Timer::Stop()`.

Here is the call graph for this function:



## 7.51 inst/include/doxygen/Examples.h File Reference

## 7.52 inst/include/Rcpp.h File Reference

```

#include <RcppCommon.h>
#include <Rcpp/RObject.h>
#include <Rcpp/S4.h>
#include <Rcpp/Reference.h>
#include <Rcpp/clone.h>
#include <Rcpp/grow.h>
#include <Rcpp/Dimension.h>
#include <Rcpp/Symbol.h>
#include <Rcpp/Environment.h>
#include <Rcpp/Vector.h>
#include <Rcpp/sugar/nona/nona.h>
#include <Rcpp/Fast.h>
#include <Rcpp/Extractor.h>
#include <Rcpp/Promise.h>
#include <Rcpp/XPtr.h>
#include <Rcpp/DottedPairImpl.h>
#include <Rcpp/Function.h>
#include <Rcpp/Language.h>
#include <Rcpp/DottedPair.h>
#include <Rcpp/Pairlist.h>
#include <Rcpp/StretchyList.h>
#include <Rcpp/WeakReference.h>
#include <Rcpp/StringTransformer.h>
#include <Rcpp/Formula.h>
#include <Rcpp/DataFrame.h>
#include <Rcpp/exceptions_impl.h>

```

```

#include <Rcpp/date_datetime/date_datetime.h>
#include <Rcpp/Na_Proxy.h>
#include <Rcpp/Module.h>
#include <Rcpp/InternalFunction.h>
#include <Rcpp/Nullable.h>
#include <Rcpp/RNGScope.h>
#include <Rcpp/sugar/sugar.h>
#include <Rcpp/stats/stats.h>
#include <Rcpp/Rmath.h>
#include <Rcpp/internal/wrap_end.h>
#include <Rcpp/platform/solaris.h>
#include <Rcpp/api/meat/meat.h>
#include <Rcpp/algorithm.h>

```

Include dependency graph for Rcpp.h:



This graph shows which files directly or indirectly include this file:



## Macros

- `#define RCPP_NEW_DATE_DATETIME_VECTORS 1`

### 7.52.1 Macro Definition Documentation

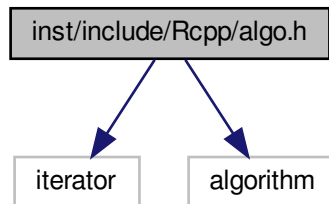
#### 7.52.1.1 RCPP\_NEW\_DATE\_DATETIME\_VECTORS

```
#define RCPP_NEW_DATE_DATETIME_VECTORS 1
```

Definition at line 62 of file Rcpp.h.

## 7.53 inst/include/Rcpp/algo.h File Reference

```
#include <iterator>
#include <algorithm>
Include dependency graph for algo.h:
```



This graph shows which files directly or indirectly include this file:



### Namespaces

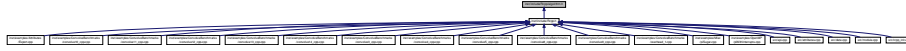
- [Rcpp](#)  
*Rcpp API.*

### Functions

- `template<class InputIterator, class T >`  
`bool Rcpp::__any (InputIterator first, InputIterator last, const T &value, std::input_iterator_tag)`
- `template<class RandomAccessIterator, class T >`  
`bool Rcpp::__any (RandomAccessIterator __first, RandomAccessIterator __last, const T &__val, std::random_↵  
access_iterator_tag)`
- `template<class InputIterator, class T >`  
`bool Rcpp::any (InputIterator first, InputIterator last, const T &value)`
- `template<class InputIterator, class Predicate >`  
`bool Rcpp::__any_if (InputIterator first, InputIterator last, Predicate pred, std::input_iterator_tag)`
- `template<class RandomAccessIterator, class Predicate >`  
`bool Rcpp::__any_if (RandomAccessIterator __first, RandomAccessIterator __last, Predicate __pred, std↵  
::random_access_iterator_tag)`
- `template<class InputIterator, class Predicate >`  
`bool Rcpp::any_if (InputIterator first, InputIterator last, Predicate pred)`

## 7.54 inst/include/Rcpp/algorithm.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::algorithm::helpers::CTYPE\\_CHAR](#)
- struct [Rcpp::algorithm::helpers::CTYPE\\_SHORT](#)
- struct [Rcpp::algorithm::helpers::CTYPE\\_INT](#)
- struct [Rcpp::algorithm::helpers::CTYPE\\_LONG](#)
- struct [Rcpp::algorithm::helpers::CTYPE\\_FLOAT](#)
- struct [Rcpp::algorithm::helpers::CTYPE\\_DOUBLE](#)
- struct [Rcpp::algorithm::helpers::CTYPE\\_LONG\\_DOUBLE](#)
- struct [Rcpp::algorithm::helpers::CTYPE\\_STRING](#)
- struct [Rcpp::algorithm::helpers::CTYPE\\_UNSIGNED\\_CHAR](#)
- struct [Rcpp::algorithm::helpers::CTYPE\\_UNSIGNED\\_SHORT](#)
- struct [Rcpp::algorithm::helpers::CTYPE\\_UNSIGNED\\_INT](#)
- struct [Rcpp::algorithm::helpers::CTYPE\\_UNSIGNED\\_LONG](#)
- struct [Rcpp::algorithm::helpers::CTYPE\\_UNKNOWN](#)
- struct [Rcpp::algorithm::helpers::ctype\\_helper< I >](#)
- struct [Rcpp::algorithm::helpers::ctype\\_helper< sizeof\(CTYPE\\_CHAR\) >](#)
- struct [Rcpp::algorithm::helpers::ctype\\_helper< sizeof\(CTYPE\\_SHORT\) >](#)
- struct [Rcpp::algorithm::helpers::ctype\\_helper< sizeof\(CTYPE\\_INT\) >](#)
- struct [Rcpp::algorithm::helpers::ctype\\_helper< sizeof\(CTYPE\\_LONG\) >](#)
- struct [Rcpp::algorithm::helpers::ctype\\_helper< sizeof\(CTYPE\\_FLOAT\) >](#)
- struct [Rcpp::algorithm::helpers::ctype\\_helper< sizeof\(CTYPE\\_DOUBLE\) >](#)
- struct [Rcpp::algorithm::helpers::ctype\\_helper< sizeof\(CTYPE\\_LONG\\_DOUBLE\) >](#)
- struct [Rcpp::algorithm::helpers::ctype\\_helper< sizeof\(CTYPE\\_STRING\) >](#)
- struct [Rcpp::algorithm::helpers::ctype\\_helper< sizeof\(CTYPE\\_UNSIGNED\\_CHAR\) >](#)
- struct [Rcpp::algorithm::helpers::ctype\\_helper< sizeof\(CTYPE\\_UNSIGNED\\_SHORT\) >](#)
- struct [Rcpp::algorithm::helpers::ctype\\_helper< sizeof\(CTYPE\\_UNSIGNED\\_INT\) >](#)
- struct [Rcpp::algorithm::helpers::ctype\\_helper< sizeof\(CTYPE\\_UNSIGNED\\_LONG\) >](#)
- struct [Rcpp::algorithm::helpers::ctype< T >](#)
- struct [Rcpp::algorithm::helpers::decays\\_to\\_ctype< T >](#)
- struct [Rcpp::algorithm::helpers::rtype\\_helper< T >](#)
- struct [Rcpp::algorithm::helpers::rtype\\_helper< double >](#)
- struct [Rcpp::algorithm::helpers::rtype\\_helper< int >](#)
- struct [Rcpp::algorithm::helpers::rtype< T >](#)
- struct [Rcpp::algorithm::helpers::log](#)
- struct [Rcpp::algorithm::helpers::exp](#)
- struct [Rcpp::algorithm::helpers::sqrt](#)

## Namespaces

- [Rcpp](#)
  - Rcpp API.*
- [Rcpp::algorithm](#)
- [Rcpp::algorithm::helpers](#)

## Macros

- `#define RCPP_CONSTEXPR_FUNC`
- `#define RCPP_CONSTEXPR_VAR const`

## Functions

- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >←`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type`  
[Rcpp::algorithm::sum](#) (InputIterator begin, InputIterator end)
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >←`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type`  
[Rcpp::algorithm::sum\\_nona](#) (InputIterator begin, InputIterator end)
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >←`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type`  
[Rcpp::algorithm::prod](#) (InputIterator begin, InputIterator end)
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >←`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type`  
[Rcpp::algorithm::prod\\_nona](#) (InputIterator begin, InputIterator end)
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >←`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type`  
[Rcpp::algorithm::max](#) (InputIterator begin, InputIterator end)
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >←`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type`  
[Rcpp::algorithm::max\\_nona](#) (InputIterator begin, InputIterator end)
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >←`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type`  
[Rcpp::algorithm::min](#) (InputIterator begin, InputIterator end)
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >←`  
`::value, typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >::type >::type`  
[Rcpp::algorithm::min\\_nona](#) (InputIterator begin, InputIterator end)
- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value`  
`&&traits::same_type< typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >←`  
`::type, double >::value, double >::type` [Rcpp::algorithm::mean](#) (InputIterator begin, InputIterator end)



- `template<typename InputIterator >`  
`traits::enable_if< helpers::decays_to_ctype< typename std::iterator_traits< InputIterator >::value_type >::value`  
`&&traits::same_type< typename helpers::ctype< typename std::iterator_traits< InputIterator >::value_type >↔`  
`::type, int >::value, double >::type` `Rcpp::algorithm::mean` (InputIterator begin, InputIterator end)
- `template<typename InputIterator , typename OutputIterator >`  
`void` `Rcpp::algorithm::log` (InputIterator begin, InputIterator end, OutputIterator out)
- `template<typename InputIterator , typename OutputIterator >`  
`void` `Rcpp::algorithm::exp` (InputIterator begin, InputIterator end, OutputIterator out)
- `template<typename InputIterator , typename OutputIterator >`  
`void` `Rcpp::algorithm::sqrt` (InputIterator begin, InputIterator end, OutputIterator out)

## 7.54.1 Macro Definition Documentation

### 7.54.1.1 RCPP\_CONSTEXPR\_FUNC

```
#define RCPP_CONSTEXPR_FUNC
```

Definition at line 29 of file algorithm.h.

### 7.54.1.2 RCPP\_CONSTEXPR\_VAR

```
#define RCPP_CONSTEXPR_VAR const
```

Definition at line 30 of file algorithm.h.

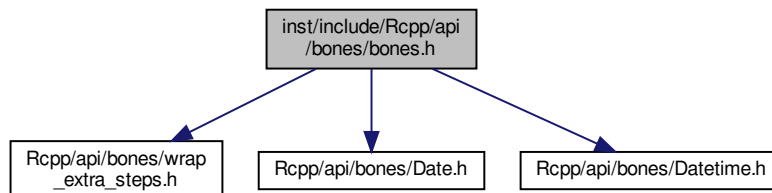
## 7.55 inst/include/Rcpp/api/bones/bones.h File Reference

```
#include <Rcpp/api/bones/wrap_extra_steps.h>
```

```
#include <Rcpp/api/bones/Date.h>
```

```
#include <Rcpp/api/bones/Datetime.h>
```

Include dependency graph for bones.h:



This graph shows which files directly or indirectly include this file:



## 7.56 inst/include/Rcpp/api/bones/Date.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::traits::wrap\\_type\\_traits< Rcpp::Date >](#)
- struct [Rcpp::traits::r\\_type\\_traits< Rcpp::Date >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, Rcpp::Date > >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< Rcpp::Date >](#)

### Namespaces

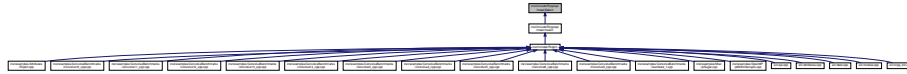
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*
- [Rcpp::internal](#)  
*internal implementation details*

### Functions

- template<> SEXP [Rcpp::wrap\\_extra\\_steps< Rcpp::Date >](#) (SEXP)
- template<> double [Rcpp::internal::caster< Rcpp::Date, double >](#) ([Rcpp::Date](#) from)
- template<> [Rcpp::Date](#) [Rcpp::internal::caster< double, Rcpp::Date >](#) (double from)

## 7.57 inst/include/Rcpp/api/meat/Date.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

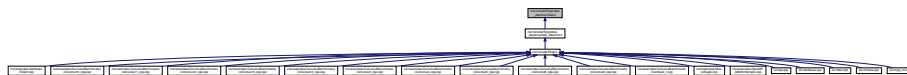
- [Rcpp](#)  
*Rcpp API.*

### Functions

- `template<> SEXP Rcpp::wrap (const Date &date)`

## 7.58 inst/include/Rcpp/date\_datetime/Date.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::Date](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Functions

- `template<> SEXP Rcpp::wrap< Rcpp::Date > (const Rcpp::Date &date)`
- `template<> double Rcpp::internal::caster< Rcpp::Date, double > (Rcpp::Date from)`
- `template<> Rcpp::Date Rcpp::internal::caster< double, Rcpp::Date > (double from)`
- `template<> SEXP Rcpp::wrap_extra_steps< Rcpp::Date > (SEXP)`
- `Date Rcpp::operator+ (const Date &date, int offset)`
- `double Rcpp::operator- (const Date &d1, const Date &d2)`
- `bool Rcpp::operator< (const Date &d1, const Date &d2)`
- `bool Rcpp::operator> (const Date &d1, const Date &d2)`
- `bool Rcpp::operator== (const Date &d1, const Date &d2)`
- `bool Rcpp::operator>= (const Date &d1, const Date &d2)`
- `bool Rcpp::operator<= (const Date &d1, const Date &d2)`
- `bool Rcpp::operator!= (const Date &d1, const Date &d2)`
- `std::ostream & Rcpp::operator<< (std::ostream &os, const Date d)`
- `SEXP Rcpp::internal::getPosixClasses ()`
- `SEXP Rcpp::internal::new_posixt_object (double d)`
- `SEXP Rcpp::internal::new_date_object (double d)`

## 7.59 inst/include/Rcpp/api/bones/Datetime.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- `struct Rcpp::traits::wrap_type_traits< Rcpp::Datetime >`
- `struct Rcpp::traits::r_type_traits< Rcpp::Datetime >`
- `struct Rcpp::traits::r_type_traits< std::pair< const std::string, Rcpp::Datetime > >`
- `struct Rcpp::traits::r_sexptype_traits< Rcpp::Datetime >`

## Namespaces

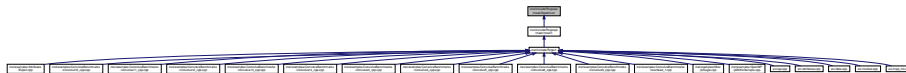
- `Rcpp`  
*Rcpp API.*
- `Rcpp::traits`  
*traits used to dispatch wrap*
- `Rcpp::internal`  
*internal implementation details*

## Functions

- `template<> SEXP Rcpp::wrap_extra_steps< Rcpp::Datetime > (SEXP)`
- `template<> double Rcpp::internal::caster< Rcpp::Datetime, double > (Rcpp::Datetime from)`
- `template<> Rcpp::Datetime Rcpp::internal::caster< double, Rcpp::Datetime > (double from)`

## 7.60 inst/include/Rcpp/api/meat/Datetime.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Functions

- `template<> SEXP Rcpp::wrap_extra_steps< Rcpp::Datetime > (SEXP)`
- `template<> SEXP Rcpp::wrap< Datetime > (const Datetime &date)`

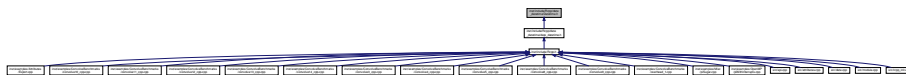
## 7.61 inst/include/Rcpp/date\_datetime/Datetime.h File Reference

```
#include <RcppCommon.h>
```

Include dependency graph for Datetime.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::Datetime](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Functions

- `template<> SEXP Rcpp::wrap< Rcpp::Datetime > (const Rcpp::Datetime &dt)`
- `template<> double Rcpp::internal::caster< Rcpp::Datetime, double > (Rcpp::Datetime from)`
- `template<> Rcpp::Datetime Rcpp::internal::caster< double, Rcpp::Datetime > (double from)`
- `template<> SEXP Rcpp::wrap_extra_steps< Rcpp::Datetime > (SEXP)`
- `Datetime Rcpp::operator+ (const Datetime &datetime, double offset)`
- `Datetime Rcpp::operator+ (const Datetime &datetime, int offset)`
- `double Rcpp::operator- (const Datetime &d1, const Datetime &d2)`
- `bool Rcpp::operator< (const Datetime &d1, const Datetime &d2)`
- `bool Rcpp::operator> (const Datetime &d1, const Datetime &d2)`
- `bool Rcpp::operator== (const Datetime &d1, const Datetime &d2)`
- `bool Rcpp::operator>= (const Datetime &d1, const Datetime &d2)`
- `bool Rcpp::operator<= (const Datetime &d1, const Datetime &d2)`
- `bool Rcpp::operator!= (const Datetime &d1, const Datetime &d2)`
- `std::ostream & Rcpp::operator<< (std::ostream &os, const Datetime d)`

## 7.62 inst/include/Rcpp/api/bones/wrap\_extra\_steps.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

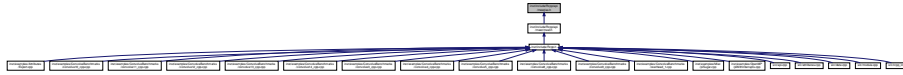
- [Rcpp](#)  
*Rcpp API.*

## Functions

- `template<typename T >`  
`SEXP Rcpp::wrap_extra_steps (SEXP x)`

## 7.63 inst/include/Rcpp/api/meat/as.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

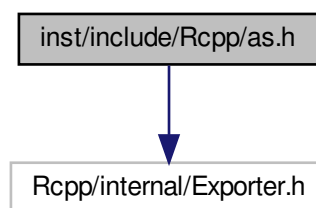
### Functions

- void \* [Rcpp::internal::as\\_module\\_object\\_internal](#) (SEXP obj)

## 7.64 inst/include/Rcpp/as.h File Reference

```
#include <Rcpp/internal/Exporter.h>
```

Include dependency graph for as.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

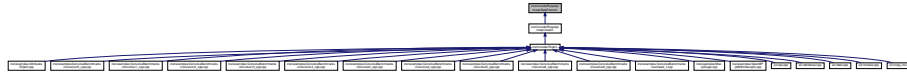
## Functions

- `template<typename T >`  
`T Rcpp::internal::primitive_as (SEXP x)`
- `template<typename T >`  
`T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_primitive_tag)`
- `const char * Rcpp::internal::check_single_string (SEXP x)`
- `template<typename T >`  
`T Rcpp::internal::as_string (SEXP x, Rcpp::traits::true_type)`
- `template<typename T >`  
`T Rcpp::internal::as_string (SEXP x, Rcpp::traits::false_type)`
- `template<typename T >`  
`T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_string_tag)`
- `template<typename T >`  
`T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_RcppString_tag)`
- `template<typename T >`  
`T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_generic_tag)`
- `void * Rcpp::internal::as_module_object_internal (SEXP obj)`
- `template<typename T >`  
`object< T > Rcpp::internal::as_module_object (SEXP x)`
- `template<typename T >`  
`T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_module_object_const_pointer_tag)`
- `template<typename T >`  
`T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_module_object_pointer_tag)`
- `template<typename T >`  
`T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_module_object_tag)`
- `template<typename T >`  
`T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_module_object_reference_tag)`
- `template<typename T >`  
`T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_module_object_const_reference_tag)`
- `template<typename T >`  
`T Rcpp::internal::as (SEXP x, ::Rcpp::traits::r_type_enum_tag)`
- `template<typename T >`  
`T Rcpp::as (SEXP x)`
- `template<> char Rcpp::as< char > (SEXP x)`
- `template<typename T >`  
`traits::remove_const_and_reference< T >::type Rcpp::bare_as (SEXP x)`
- `template<> SEXP Rcpp::as (SEXP x)`



## 7.65 inst/include/Rcpp/api/meat/DataFrame.h File Reference

This graph shows which files directly or indirectly include this file:



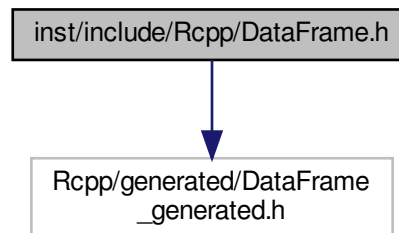
### Namespaces

- [Rcpp](#)  
*Rcpp API.*

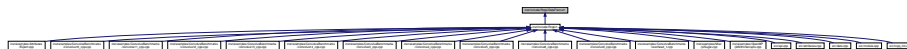
## 7.66 inst/include/Rcpp/DataFrame.h File Reference

```
#include <Rcpp/generated/DataFrame_generated.h>
```

Include dependency graph for DataFrame.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::DataFrame\\_Impl< StoragePolicy >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Typedefs

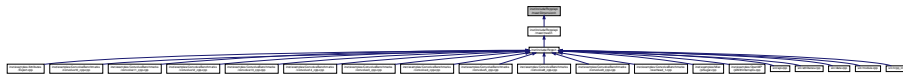
- typedef `DataFrame_Impl< PreserveStorage >` [Rcpp::DataFrame](#)

## Functions

- SEXP [Rcpp::internal::empty\\_data\\_frame](#) ()

## 7.67 inst/include/Rcpp/api/meat/Dimension.h File Reference

This graph shows which files directly or indirectly include this file:

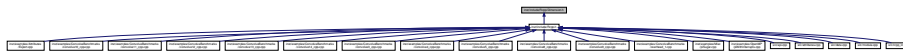


## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.68 inst/include/Rcpp/Dimension.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

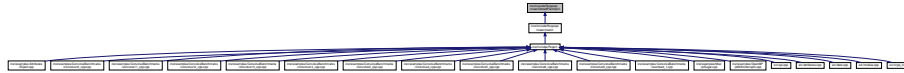
- class [Rcpp::Dimension](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.69 inst/include/Rcpp/api/meat/DottedPairImpl.h File Reference

This graph shows which files directly or indirectly include this file:

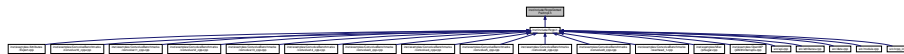


## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.70 inst/include/Rcpp/DottedPairImpl.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::DottedPairImpl< CLASS >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.71 inst/include/Rcpp/api/meat/Environment.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

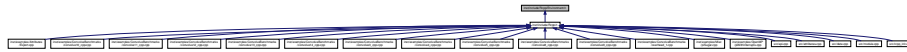
- [Rcpp](#)  
*Rcpp API.*

## Functions

- Environment [Rcpp::new\\_env](#) (int size=29)
- Environment [Rcpp::new\\_env](#) (SEXP parent, int size=29)

## 7.72 inst/include/Rcpp/Environment.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Typedefs

- typedef Environment\_Impl< PreserveStorage > [Rcpp::Environment](#)

## Functions

- [Rcpp::RCPP\\_API\\_CLASS](#) (Environment\_Impl)
- [Rcpp::Environment\\_Impl](#) (SEXP x)
- [Rcpp::Environment\\_Impl](#) (const std::string &name)
- [Rcpp::Environment\\_Impl](#) (int pos)
- SEXP [Rcpp::ls](#) (bool all) const
- SEXP [Rcpp::get](#) (const std::string &name) const
- SEXP [Rcpp::get](#) (Symbol name) const
- SEXP [Rcpp::find](#) (const std::string &name) const
- SEXP [Rcpp::find](#) (Symbol name) const
- bool [Rcpp::exists](#) (const std::string &name) const
- bool [Rcpp::assign](#) (const std::string &name, SEXP x) const
- bool [Rcpp::assign](#) (const std::string &name, const Shield< SEXP > &x) const
- template<typename WRAPPABLE >  
bool [Rcpp::assign](#) (const std::string &name, const WRAPPABLE &x) const
- bool [Rcpp::isLocked](#) () const

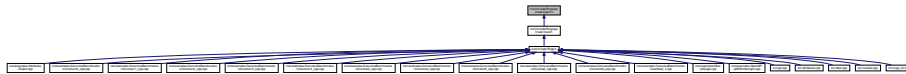
- bool [Rcpp::remove](#) (const std::string &name)
- void [Rcpp::lock](#) (bool bindings=false)
- void [Rcpp::lockBinding](#) (const std::string &name)
- void [Rcpp::unlockBinding](#) (const std::string &name)
- bool [Rcpp::bindingsLocked](#) (const std::string &name) const
- bool [Rcpp::bindingsActive](#) (const std::string &name) const
- bool [Rcpp::is\\_user\\_database](#) () const
- static Environment\_Impl [Rcpp::global\\_env](#) ()
- static Environment\_Impl [Rcpp::empty\\_env](#) ()
- static Environment\_Impl [Rcpp::base\\_env](#) ()
- static Environment\_Impl [Rcpp::base\\_namespace](#) ()
- static Environment\_Impl [Rcpp::Rcpp\\_namespace](#) ()
- static Environment\_Impl [Rcpp::namespace\\_env](#) (const std::string &package)
- Environment\_Impl [Rcpp::parent](#) () const
- Environment\_Impl [Rcpp::new\\_child](#) (bool hashed) const
- void [Rcpp::update](#) (SEXP)

## Variables

- public [Rcpp::BindingPolicy](#)< Environment\_Impl< StoragePolicy > >

## 7.73 inst/include/Rcpp/api/meat/export.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::traits::ContainerExporter](#)< ContainerTemplate, T >
- struct [Rcpp::traits::container\\_exporter](#)< Container, int >
- struct [Rcpp::traits::container\\_exporter](#)< Container, double >

## Namespaces

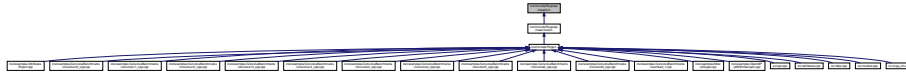
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## Functions

- `template<typename InputIterator, typename value_type >`  
`void Rcpp::internal::export_range__dispatch (SEXP x, InputIterator first, ::Rcpp::traits::r_type_generic_tag)`

## 7.74 inst/include/Rcpp/api/meat/is.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Functions

- `bool Rcpp::internal::is_atomic (SEXP x)`
- `bool Rcpp::internal::is_matrix (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< int > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< double > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< bool > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< std::string > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< String > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< Rcomplex > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< CharacterVector > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< CharacterMatrix > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< RObject > (SEXP)`
- `template<> bool Rcpp::internal::is__simple< IntegerVector > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< ComplexVector > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< RawVector > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< NumericVector > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< LogicalVector > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< Language > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< DottedPair > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< List > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< IntegerMatrix > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< ComplexMatrix > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< RawMatrix > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< NumericMatrix > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< LogicalMatrix > (SEXP x)`

- `template<> bool Rcpp::internal::is__simple< GenericMatrix > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< DataFrame > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< WeakReference > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< Symbol > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< S4 > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< Reference > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< Promise > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< Pairlist > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< Function > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< Environment > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< Formula > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< Date > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< Datetime > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< DateVector > (SEXP x)`
- `template<> bool Rcpp::internal::is__simple< DatetimeVector > (SEXP x)`
- `bool Rcpp::internal::is__module_object_internal (SEXP obj, const char *clazz)`
- `template<typename T >`  
`bool Rcpp::internal::is__module__object (SEXP x)`

## 7.75 inst/include/Rcpp/is.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

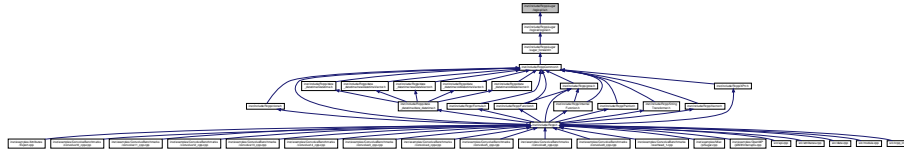
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

### Functions

- `template<typename T >`  
`bool Rcpp::internal::is__simple (SEXP x)`
- `template<typename T >`  
`bool Rcpp::internal::is__module__object (SEXP x)`
- `template<typename T >`  
`bool Rcpp::internal::is__dispatch (SEXP x, Rcpp::traits::false_type)`
- `template<typename T >`  
`bool Rcpp::internal::is__dispatch (SEXP x, Rcpp::traits::true_type)`
- `template<typename T >`  
`bool Rcpp::is (SEXP x)`

## 7.76 inst/include/Rcpp/sugar/logical/is.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [Rcpp](#)  
*Rcpp API.*

### Functions

- `template<bool NA, typename T >`  
`bool Rcpp::is\_true (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)`
- `template<bool NA, typename T >`  
`bool Rcpp::is\_false (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)`
- `template<bool NA, typename T >`  
`bool Rcpp::is\_na (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)`

## 7.77 inst/include/Rcpp/api/meat/meat.h File Reference

```
#include <Rcpp/api/meat/Rcpp_eval.h>
#include <Rcpp/api/meat/Dimension.h>
#include <Rcpp/api/meat/Date.h>
#include <Rcpp/api/meat/Datetime.h>
#include <Rcpp/api/meat/DataFrame.h>
#include <Rcpp/api/meat/S4.h>
#include <Rcpp/api/meat/Environment.h>
#include <Rcpp/api/meat/proxy.h>
#include <Rcpp/api/meat/DottedPairImpl.h>
#include <Rcpp/api/meat/StretchyList.h>
#include <Rcpp/api/meat/Vector.h>
#include <Rcpp/api/meat/is.h>
#include <Rcpp/api/meat/as.h>
#include <Rcpp/api/meat/export.h>
#include <Rcpp/api/meat/protection.h>
#include <Rcpp/api/meat/wrap.h>
#include <Rcpp/api/meat/message.h>
```

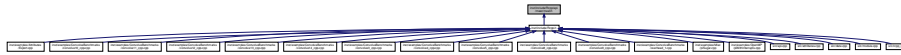


```
#include <Rcpp/api/meat/module/Module.h>
```

Include dependency graph for meat.h:

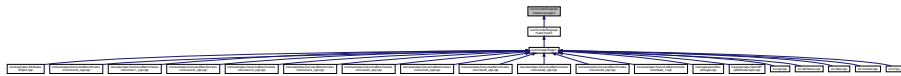


This graph shows which files directly or indirectly include this file:



## 7.78 inst/include/Rcpp/api/meat/message.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

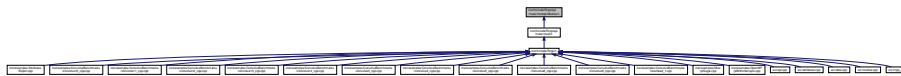
- [Rcpp](#)  
*Rcpp API.*

### Functions

- void [Rcpp::message](#) (SEXP s)

## 7.79 inst/include/Rcpp/api/meat/module/Module.h File Reference

This graph shows which files directly or indirectly include this file:

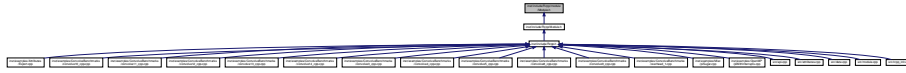


### Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.80 inst/include/Rcpp/module/Module.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::Module](#)

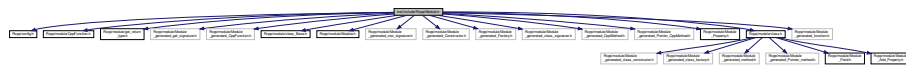
### Namespaces

- [Rcpp](#)  
*Rcpp API.*

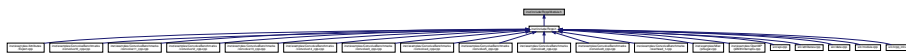
## 7.81 inst/include/Rcpp/Module.h File Reference

```
#include <Rcpp/config.h>
#include <Rcpp/module/CppFunction.h>
#include <Rcpp/module/get_return_type.h>
#include <Rcpp/module/Module_generated_get_signature.h>
#include <Rcpp/module/Module_generated_CppFunction.h>
#include <Rcpp/module/class_Base.h>
#include <Rcpp/module/Module.h>
#include <Rcpp/module/Module_generated_ctor_signature.h>
#include <Rcpp/module/Module_generated_Constructor.h>
#include <Rcpp/module/Module_generated_Factory.h>
#include <Rcpp/module/Module_generated_class_signature.h>
#include <Rcpp/module/Module_generated_CppMethod.h>
#include <Rcpp/module/Module_generated_Pointer_CppMethod.h>
#include <Rcpp/module/Module_Property.h>
#include <Rcpp/module/class.h>
#include <Rcpp/module/Module_generated_function.h>
```

Include dependency graph for Module.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::result< T >](#)
- class [Rcpp::object< T >](#)
- class [Rcpp::CppMethod< Class >](#)
- class [Rcpp::CppMethod< Class, Parent >](#)
- class [Rcpp::SignedConstructor< Class >](#)
- class [Rcpp::SignedFactory< Class >](#)
- class [Rcpp::SignedMethod< Class >](#)
- class [Rcpp::S4\\_CppConstructor< Class >](#)
- class [Rcpp::S4\\_CppOverloadedMethods< Class >](#)
- class [Rcpp::CppMethodProperty< Class >](#)
- class [Rcpp::CppMethodProperty< Class, Parent >](#)
- class [Rcpp::CppMethodFinalizer< Class >](#)
- class [Rcpp::FunctionFinalizer< Class >](#)
- class [Rcpp::S4\\_field< Class >](#)
- class [Rcpp::enum\\_< Enum, Parent >](#)
- class [Rcpp::CppClass](#)
- class [Rcpp::CppObject](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Macros

- `#define RCPP_MODULE_BOOT(name) _rcpp_module_boot_##name`
- `#define RCPP_MODULE(name)`
- `#define VARIABLE_IS_NOT_USED`
- `#define GET_MODULE_SYM ( moduleSym == NULL ? moduleSym = Rf_install("Module") : moduleSym )`
- `#define LOAD_RCPP_MODULE(NAME)`

## Typedefs

- `typedef bool(* Rcpp::ValidConstructor) (SEXP *, int)`
- `typedef bool(* Rcpp::ValidMethod) (SEXP *, int)`

## Functions

- `template<typename FROM , typename TO >`  
`std::string Rcpp::internal::get\_converter\_name (const char *from, const char *to)`
- `template<typename Class >`  
`SEXP Rcpp::internal::make\_new\_object (Class *ptr)`
- `template<typename FROM , typename TO >`  
`void Rcpp::converter (const char *from, const char *to, TO(*fun)(FROM), const char *docstring=0)`

## Variables

- static `VARIABLE_IS_NOT_USED` SEXP `moduleSym` = NULL

### 7.81.1 Macro Definition Documentation

#### 7.81.1.1 GET\_MODULE\_SYM

```
#define GET_MODULE_SYM ( moduleSym == NULL ? moduleSym = Rf_install("Module") : moduleSym )
```

Definition at line 452 of file Module.h.

#### 7.81.1.2 LOAD\_RCPP\_MODULE

```
#define LOAD_RCPP_MODULE(  
    NAME )
```

##### Value:

```
Shield<SEXP> __load_module_call__( Rf_lang2( GET_MODULE_SYM, _rcpp_module_boot_##NAME() ) ); \  
Rcpp_fast_eval( __load_module_call__, R_GlobalEnv );
```

Definition at line 455 of file Module.h.

#### 7.81.1.3 RCPP\_MODULE

```
#define RCPP_MODULE(  
    name )
```

##### Value:

```
void _rcpp_module_##name##_init() ;  
static Rcpp::Module _rcpp_module_##name( # name ) ;  
extern "C" SEXP _rcpp_module_boot_##name(){  
    ::setCurrentScope( & _rcpp_module_##name ) ;  
    _rcpp_module_##name##_init() ;  
    Rcpp::XPtr<Rcpp::Module> mod_xp(& _rcpp_module_##name , false);  
    ::setCurrentScope( 0 ) ;  
    return mod_xp ;  
}  
void _rcpp_module_##name##_init()
```

Definition at line 428 of file Module.h.

### 7.81.1.4 RCPP\_MODULE\_BOOT

```
#define RCPP_MODULE_BOOT(  
    name ) _rcpp_module_boot_##name
```

Definition at line 426 of file Module.h.

### 7.81.1.5 VARIABLE\_IS\_NOT\_USED

```
#define VARIABLE_IS_NOT_USED
```

Definition at line 444 of file Module.h.

## 7.81.2 Variable Documentation

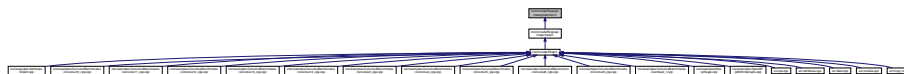
### 7.81.2.1 moduleSym

```
VARIABLE_IS_NOT_USED SEXP moduleSym = NULL [static]
```

Definition at line 448 of file Module.h.

## 7.82 inst/include/Rcpp/api/meat/protection.h File Reference

This graph shows which files directly or indirectly include this file:



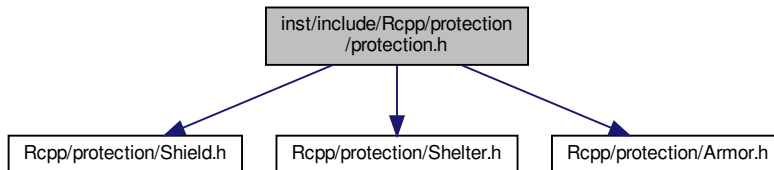
## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.83 inst/include/Rcpp/protection/protection.h File Reference

```
#include <Rcpp/protection/Shield.h>
#include <Rcpp/protection/Shelter.h>
#include <Rcpp/protection/Armor.h>
```

Include dependency graph for protection.h:

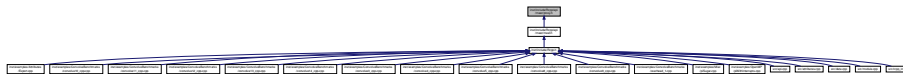


This graph shows which files directly or indirectly include this file:



## 7.84 inst/include/Rcpp/api/meat/proxy.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.85 inst/include/Rcpp/proxy/proxy.h File Reference

```
#include <Rcpp/proxy/GenericProxy.h>
#include <Rcpp/proxy/NamesProxy.h>
#include <Rcpp/proxy/RObjectMethods.h>
#include <Rcpp/proxy/AttributeProxy.h>
#include <Rcpp/proxy/TagProxy.h>
#include <Rcpp/proxy/ProtectedProxy.h>
#include <Rcpp/proxy/SlotProxy.h>
#include <Rcpp/proxy/Binding.h>
#include <Rcpp/proxy/FieldProxy.h>
#include <Rcpp/proxy/DottedPairProxy.h>
```

Include dependency graph for proxy.h:

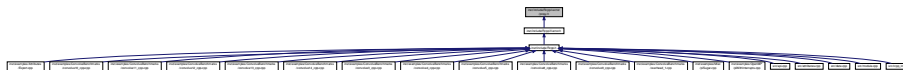


This graph shows which files directly or indirectly include this file:



## 7.86 inst/include/Rcpp/vector/proxy.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class `Rcpp::internal::simple_name_proxy< RTYPE, StoragePolicy >`
- class `Rcpp::internal::string_name_proxy< RTYPE, StoragePolicy >`
- class `Rcpp::internal::generic_name_proxy< RTYPE, StoragePolicy >`
- struct `Rcpp::traits::r_vector_name_proxy< RTYPE, StoragePolicy >`
- struct `Rcpp::traits::r_vector_name_proxy< STRSXP, StoragePolicy >`
- struct `Rcpp::traits::r_vector_name_proxy< VECSXP, StoragePolicy >`
- struct `Rcpp::traits::r_vector_name_proxy< EXPRSXP, StoragePolicy >`
- struct `Rcpp::traits::r_vector_proxy< RTYPE, StoragePolicy >`
- struct `Rcpp::traits::r_vector_proxy< STRSXP, StoragePolicy >`
- struct `Rcpp::traits::r_vector_proxy< EXPRSXP, StoragePolicy >`

- struct [Rcpp::traits::r\\_vector\\_proxy< VECSXP, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_const\\_proxy< RTYPE, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_const\\_proxy< STRSXP, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_const\\_proxy< VECSXP, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_const\\_proxy< EXPRSXP, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_iterator< RTYPE, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_const\\_iterator< RTYPE, StoragePolicy >](#)
- struct [Rcpp::traits::proxy\\_based\\_iterator< RTYPE, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_iterator< VECSXP, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_iterator< EXPRSXP, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_iterator< STRSXP, StoragePolicy >](#)
- struct [Rcpp::traits::proxy\\_based\\_const\\_iterator< RTYPE, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_const\\_iterator< VECSXP, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_const\\_iterator< EXPRSXP, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_const\\_iterator< STRSXP, StoragePolicy >](#)

## Namespaces

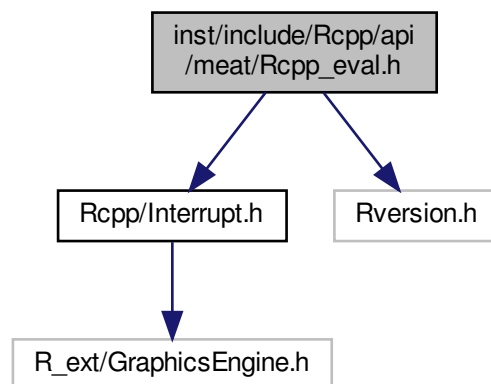
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.87 inst/include/Rcpp/api/meat/Rcpp\_eval.h File Reference

```
#include <Rcpp/Interrupt.h>
```

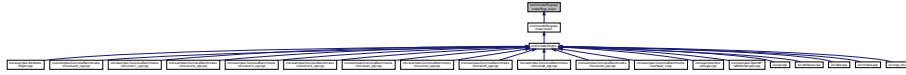
```
#include <Rversion.h>
```

Include dependency graph for Rcpp\_eval.h:





This graph shows which files directly or indirectly include this file:



## Namespaces

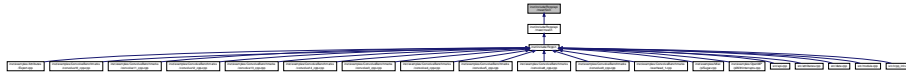
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Functions

- SEXP [Rcpp::internal::Rcpp\\_eval\\_impl](#) (SEXP expr, SEXP env)
- SEXP [Rcpp::Rcpp\\_fast\\_eval](#) (SEXP expr, SEXP env)
- SEXP [Rcpp::Rcpp\\_eval](#) (SEXP expr, SEXP env)

## 7.88 inst/include/Rcpp/api/meat/S4.h File Reference

This graph shows which files directly or indirectly include this file:

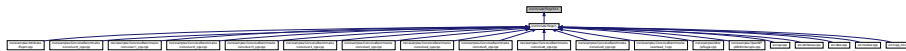


## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.89 inst/include/Rcpp/S4.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Typedefs

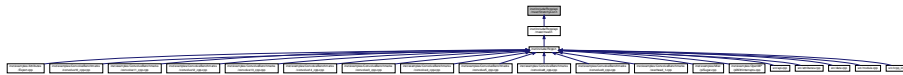
- typedef S4\_Impl< PreserveStorage > [Rcpp::S4](#)

## Functions

- [Rcpp::RCPP\\_API\\_CLASS](#) (S4\_Impl)

## 7.90 inst/include/Rcpp/api/meat/StretchyList.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.91 inst/include/Rcpp/StretchyList.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Typedefs

- typedef StretchyList\_Impl< PreserveStorage > [Rcpp::StretchyList](#)

## Functions

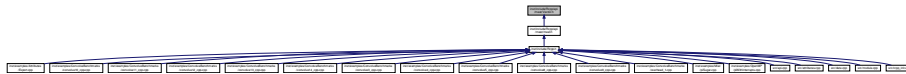
- [Rcpp::RCPP\\_API\\_CLASS](#) (StretchyList\_Impl)
- [Rcpp::StretchyList\\_Impl](#) ()
- [Rcpp::StretchyList\\_Impl](#) (SEXP x)
- void [Rcpp::update](#) (SEXP)
- [Rcpp::operator SEXP](#) () const
- template<typename T >  
StretchyList\_Impl & [Rcpp::push\\_back](#) (const T &obj)
- template<typename T >  
StretchyList\_Impl & [Rcpp::push\\_front](#) (const T &obj)
- template<typename T >  
StretchyList\_Impl & [Rcpp::push\\_back\\_\\_impl](#) (const T &obj, traits::true\_type)
- template<typename T >  
StretchyList\_Impl & [Rcpp::push\\_back\\_\\_impl](#) (const T &obj, traits::false\_type)
- template<typename T >  
StretchyList\_Impl & [Rcpp::push\\_front\\_\\_impl](#) (const T &obj, traits::true\_type)
- template<typename T >  
StretchyList\_Impl & [Rcpp::push\\_front\\_\\_impl](#) (const T &obj, traits::false\_type)

## Variables

- public [Rcpp::DottedPairProxyPolicy](#)< StretchyList\_Impl< StoragePolicy > >

## 7.92 inst/include/Rcpp/api/meat/Vector.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

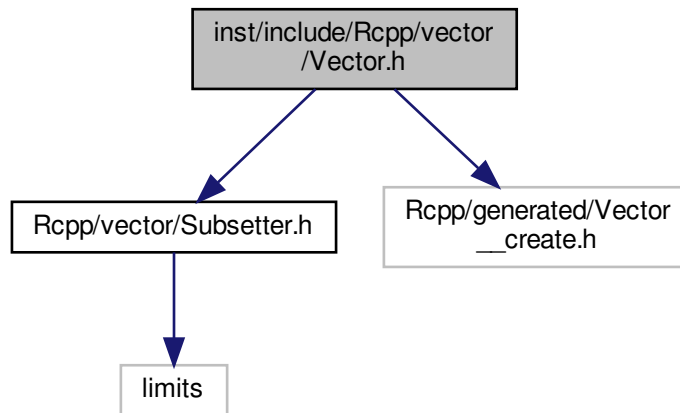
## Functions

- `template<typename T >`  
SEXP `Rcpp::internal::wrap_range_sugar_expression` (const T &object, `Rcpp::traits::true_type`)

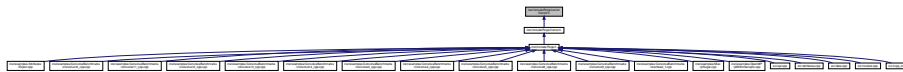
## 7.93 inst/include/Rcpp/vector/Vector.h File Reference

```
#include <Rcpp/vector/Subsetter.h>
#include <Rcpp/generated/Vector__create.h>
```

Include dependency graph for Vector.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `Rcpp::Vector< RTYPE, StoragePolicy >`

## Namespaces

- `Rcpp`  
*Rcpp API.*

## Functions

- `template<int RTYPE, template< class > class StoragePolicy>`  
`std::ostream & Rcpp::operator<< (std::ostream &s, const Vector< RTYPE, StoragePolicy > &rhs)`
- `template<template< class > class StoragePolicy>`  
`std::ostream & Rcpp::operator<< (std::ostream &s, const Vector< STRSXP, StoragePolicy > &rhs)`

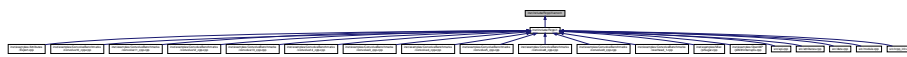
## 7.94 inst/include/Rcpp/Vector.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/exceptions.h>
#include <Rcpp/r_cast.h>
#include <Rcpp/vector/00_forward_Vector.h>
#include <Rcpp/vector/no_init.h>
#include <Rcpp/vector/00_forward_proxy.h>
#include <Rcpp/vector/vector_from_string.h>
#include <Rcpp/vector/converter.h>
#include <Rcpp/vector/RangeIndexer.h>
#include <Rcpp/vector/Vector.h>
#include <Rcpp/vector/proxy.h>
#include <Rcpp/vector/traits.h>
#include <Rcpp/vector/DimNameProxy.h>
#include <Rcpp/vector/Matrix.h>
#include <Rcpp/vector/SubMatrix.h>
#include <Rcpp/vector/MatrixRow.h>
#include <Rcpp/vector/MatrixColumn.h>
#include <Rcpp/vector/instantiation.h>
#include <Rcpp/vector/string_proxy.h>
#include <Rcpp/vector/const_string_proxy.h>
#include <Rcpp/vector/generic_proxy.h>
#include <Rcpp/vector/const_generic_proxy.h>
#include <Rcpp/String.h>
#include <Rcpp/vector/LazyVector.h>
#include <Rcpp/vector/swap.h>
#include <Rcpp/vector/ChildVector.h>
#include <Rcpp/vector/ListOf.h>
```

Include dependency graph for Vector.h:



This graph shows which files directly or indirectly include this file:

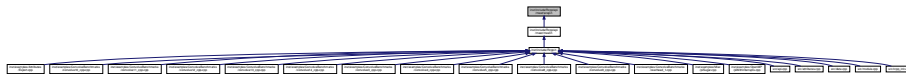


## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.95 inst/include/Rcpp/api/meat/wrap.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Functions

- `template<typename InputIterator , typename KEY , typename VALUE , int RTYPE>`  
SEXP [Rcpp::internal::range\\_wrap\\_dispatch\\_\\_impl\\_\\_pair](#) (InputIterator first, InputIterator last, [Rcpp::traits::true\\_type](#))
- `template<typename InputIterator , typename KEY , typename VALUE , int RTYPE>`  
SEXP [Rcpp::internal::range\\_wrap\\_dispatch\\_\\_impl\\_\\_pair](#) (InputIterator first, InputIterator last, [Rcpp::traits::false\\_type](#))

## 7.96 inst/include/Rcpp/barrier.h File Reference

This graph shows which files directly or indirectly include this file:



## Functions

- SEXP [get\\_string\\_elt](#) (SEXP, R\_xlen\_t)
- const char \* [char\\_get\\_string\\_elt](#) (SEXP, R\_xlen\_t)
- void [set\\_string\\_elt](#) (SEXP, R\_xlen\_t, SEXP)
- void [char\\_set\\_string\\_elt](#) (SEXP, R\_xlen\_t, const char \*)
- SEXP \* [get\\_string\\_ptr](#) (SEXP)
- SEXP [get\\_vector\\_elt](#) (SEXP, R\_xlen\_t)
- void [set\\_vector\\_elt](#) (SEXP, R\_xlen\_t, SEXP)
- SEXP \* [get\\_vector\\_ptr](#) (SEXP)
- const char \* [char\\_nocheck](#) (SEXP)
- void \* [dataptr](#) (SEXP)

### 7.96.1 Function Documentation

#### 7.96.1.1 [char\\_get\\_string\\_elt\(\)](#)

```
const char* char_get_string_elt (  
    SEXP s,  
    R_xlen_t i ) [inline]
```

Definition at line 216 of file routines.h.

References [GET\\_CALLABLE](#).

Referenced by [registerFunctions\(\)](#).

#### 7.96.1.2 [char\\_nocheck\(\)](#)

```
const char* char_nocheck (  
    SEXP x ) [inline]
```

Definition at line 258 of file routines.h.

References [GET\\_CALLABLE](#).

Referenced by [registerFunctions\(\)](#), and [Rcpp::String::setBuffer\(\)](#).

### 7.96.1.3 char\_set\_string\_elt()

```
void char_set_string_elt (
    SEXP s,
    R_xlen_t i,
    const char * v ) [inline]
```

Definition at line 228 of file routines.h.

References GET\_CALLABLE.

Referenced by registerFunctions().

### 7.96.1.4 dataptr()

```
void* dataptr (
    SEXP x ) [inline]
```

Definition at line 264 of file routines.h.

References GET\_CALLABLE.

Referenced by registerFunctions().

### 7.96.1.5 get\_string\_elt()

```
SEXP get_string_elt (
    SEXP s,
    R_xlen_t i ) [inline]
```

Definition at line 210 of file routines.h.

References GET\_CALLABLE.

Referenced by registerFunctions().

### 7.96.1.6 get\_string\_ptr()

```
SEXP* get_string_ptr (
    SEXP s ) [inline]
```

Definition at line 234 of file routines.h.

References GET\_CALLABLE.

Referenced by Rcpp::sugar::get\_const\_begin(), and registerFunctions().



### 7.96.1.7 `get_vector_elt()`

```
SEXP get_vector_elt (
    SEXP v,
    R_xlen_t i ) [inline]
```

Definition at line 240 of file routines.h.

References `GET_CALLABLE`.

Referenced by `registerFunctions()`.

### 7.96.1.8 `get_vector_ptr()`

```
SEXP* get_vector_ptr (
    SEXP v ) [inline]
```

Definition at line 252 of file routines.h.

References `GET_CALLABLE`.

Referenced by `registerFunctions()`.

### 7.96.1.9 `set_string_elt()`

```
void set_string_elt (
    SEXP s,
    R_xlen_t i,
    SEXP v ) [inline]
```

Definition at line 222 of file routines.h.

References `GET_CALLABLE`.

Referenced by `registerFunctions()`.

### 7.96.1.10 `set_vector_elt()`

```
void set_vector_elt (
    SEXP v,
    R_xlen_t i,
    SEXP x ) [inline]
```

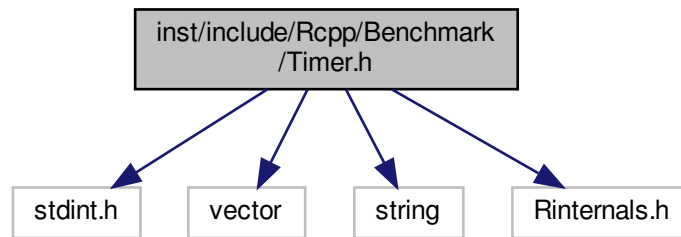
Definition at line 246 of file routines.h.

References `GET_CALLABLE`.

Referenced by `registerFunctions()`.

## 7.97 inst/include/Rcpp/Benchmark/Timer.h File Reference

```
#include <stdint.h>
#include <vector>
#include <string>
#include <Rinternals.h>
Include dependency graph for Timer.h:
```



### Classes

- class [Rcpp::Timer](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*

### Macros

- #define [R\\_NO\\_REMAP](#)

### Typedefs

- typedef uint64\_t [Rcpp::nanotime\\_t](#)

### 7.97.1 Macro Definition Documentation

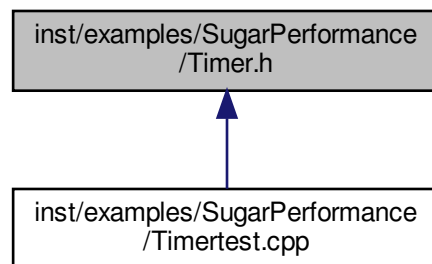
### 7.97.1.1 R\_NO\_REMAP

```
#define R_NO_REMAP
```

Definition at line 29 of file Timer.h.

## 7.98 inst/examples/SugarPerformance/Timer.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Timer](#)

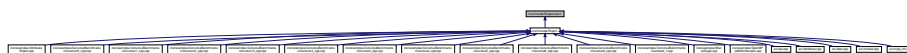
## 7.99 inst/include/Rcpp/clone.h File Reference

```
#include <RcppCommon.h>
```

Include dependency graph for clone.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Functions

- `template<typename T >`  
`T Rcpp::clone (const T &object)`

## 7.100 inst/include/Rcpp/complex.h File Reference

This graph shows which files directly or indirectly include this file:



## Macros

- `#define dplyr\_tools\_complex\_H`

## Functions

- `Rcomplex operator\* (const Rcomplex &lhs, const Rcomplex &rhs)`
- `Rcomplex operator+ (const Rcomplex &lhs, const Rcomplex &rhs)`
- `Rcomplex operator- (const Rcomplex &lhs, const Rcomplex &rhs)`
- `Rcomplex operator/ (const Rcomplex &a, const Rcomplex &b)`
- `bool operator== (const Rcomplex &a, const Rcomplex &b)`
- `std::ostream & operator<< (std::ostream &os, const Rcomplex &cplx)`

### 7.100.1 Macro Definition Documentation

#### 7.100.1.1 `dplyr_tools_complex_H`

```
#define dplyr_tools_complex_H
```

Definition at line 74 of file `complex.h`.

## 7.100.2 Function Documentation

### 7.100.2.1 `operator*()`

```
Rcomplex operator* (  
    const Rcomplex & lhs,  
    const Rcomplex & rhs ) [inline]
```

Definition at line 25 of file complex.h.

### 7.100.2.2 `operator+()`

```
Rcomplex operator+ (  
    const Rcomplex & lhs,  
    const Rcomplex & rhs ) [inline]
```

Definition at line 32 of file complex.h.

### 7.100.2.3 `operator-()`

```
Rcomplex operator- (  
    const Rcomplex & lhs,  
    const Rcomplex & rhs ) [inline]
```

Definition at line 39 of file complex.h.

### 7.100.2.4 `operator/()`

```
Rcomplex operator/ (  
    const Rcomplex & a,  
    const Rcomplex & b ) [inline]
```

Definition at line 46 of file complex.h.

### 7.100.2.5 operator<<()

```
std::ostream& operator<< (
    std::ostream & os,
    const Rcomplex & cplx ) [inline]
```

Definition at line 76 of file complex.h.

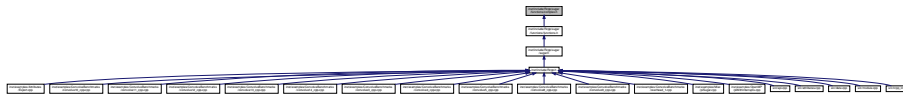
### 7.100.2.6 operator==( )

```
bool operator== (
    const Rcomplex & a,
    const Rcomplex & b ) [inline]
```

Definition at line 68 of file complex.h.

## 7.101 inst/include/Rcpp/sugar/functions/complex.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::SugarComplex< NA, RESULT\\_TYPE, T, FunPtr >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)
- [Rcpp::internal](#)  
*internal implementation details*

### Macros

- #define [RCPP\\_SUGAR\\_COMPLEX\(\\_\\_NAME\\_\\_, \\_\\_OUT\\_\\_\)](#)

## Functions

- double [Rcpp::internal::complex\\_\\_Re](#) (Rcomplex x)
- double [Rcpp::internal::complex\\_\\_Im](#) (Rcomplex x)
- double [Rcpp::internal::complex\\_\\_Mod](#) (Rcomplex x)
- Rcomplex [Rcpp::internal::complex\\_\\_Conj](#) (Rcomplex x)
- double [Rcpp::internal::complex\\_\\_Arg](#) (Rcomplex x)
- Rcomplex [Rcpp::internal::complex\\_\\_exp](#) (Rcomplex x)
- Rcomplex [Rcpp::internal::complex\\_\\_log](#) (Rcomplex x)
- Rcomplex [Rcpp::internal::complex\\_\\_sqrt](#) (Rcomplex z)
- Rcomplex [Rcpp::internal::complex\\_\\_cos](#) (Rcomplex z)
- Rcomplex [Rcpp::internal::complex\\_\\_cosh](#) (Rcomplex z)
- Rcomplex [Rcpp::internal::complex\\_\\_sin](#) (Rcomplex z)
- Rcomplex [Rcpp::internal::complex\\_\\_tan](#) (Rcomplex z)
- Rcomplex [Rcpp::internal::complex\\_\\_asin](#) (Rcomplex z)
- Rcomplex [Rcpp::internal::complex\\_\\_acos](#) (Rcomplex z)
- Rcomplex [Rcpp::internal::complex\\_\\_atan](#) (Rcomplex z)
- Rcomplex [Rcpp::internal::complex\\_\\_acosh](#) (Rcomplex z)
- Rcomplex [Rcpp::internal::complex\\_\\_asinh](#) (Rcomplex z)
- Rcomplex [Rcpp::internal::complex\\_\\_atanh](#) (Rcomplex z)
- Rcomplex [Rcpp::internal::complex\\_\\_sinh](#) (Rcomplex z)
- Rcomplex [Rcpp::internal::complex\\_\\_tanh](#) (Rcomplex z)

### 7.101.1 Macro Definition Documentation

#### 7.101.1.1 RCPP\_SUGAR\_COMPLEX

```
#define RCPP_SUGAR_COMPLEX(
    __NAME__,
    __OUT__ )
```

#### Value:

```
template <bool NA, typename T>
inline sugar::SugarComplex<NA,__OUT__,T, __OUT__ (*) (Rcomplex) >
__NAME__(const VectorBase<CPLX_SXP,NA,T>& t) {
    return sugar::SugarComplex<NA,__OUT__,T, __OUT__ (*) (Rcomplex) >(
        internal::complex_##__NAME__, t);
}
```

Definition at line 231 of file complex.h.

## 7.102 inst/include/Rcpp/config.h File Reference

This graph shows which files directly or indirectly include this file:



## Macros

- #define `Rcpp_Version`(v, p, s)  $((v) * 65536) + ((p) * 256) + (s)$
- #define `RcppDevVersion`(maj, min, rev, dev)  $((maj)*1000000) + ((min)*10000) + ((rev)*100) + (dev)$
- #define `RCPP_VERSION` `Rcpp_Version`(1,0,9)
- #define `RCPP_VERSION_STRING` "1.0.9"
- #define `RCPP_DEV_VERSION` `RcppDevVersion`(1,0,9,0)
- #define `RCPP_DEV_VERSION_STRING` "1.0.9.0"

### 7.102.1 Macro Definition Documentation

#### 7.102.1.1 RCPP\_DEV\_VERSION

```
#define RCPP_DEV_VERSION RcppDevVersion(1,0,9,0)
```

Definition at line 33 of file config.h.

#### 7.102.1.2 RCPP\_DEV\_VERSION\_STRING

```
#define RCPP_DEV_VERSION_STRING "1.0.9.0"
```

Definition at line 34 of file config.h.

#### 7.102.1.3 Rcpp\_Version

```
#define Rcpp_Version(  
    v,  
    p,  
    s )  $((v) * 65536) + ((p) * 256) + (s)$ 
```

Definition at line 24 of file config.h.

#### 7.102.1.4 RCPP\_VERSION

```
#define RCPP_VERSION Rcpp_Version(1,0,9)
```

Definition at line 29 of file config.h.



### 7.102.1.5 RCPP\_VERSION\_STRING

```
#define RCPP_VERSION_STRING "1.0.9"
```

Definition at line 30 of file config.h.

### 7.102.1.6 RcppDevVersion

```
#define RcppDevVersion(  
    maj,  
    min,  
    rev,  
    dev ) (((maj)*1000000) + ((min)*10000) + ((rev)*100) + (dev))
```

Definition at line 26 of file config.h.

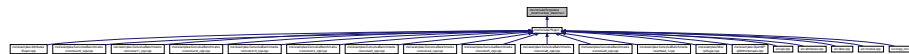
## 7.103 inst/include/Rcpp/date\_datetime/date\_datetime.h File Reference

```
#include <Rcpp/date_datetime/Date.h>  
#include <Rcpp/date_datetime/oldDateVector.h>  
#include <Rcpp/date_datetime/newDateVector.h>  
#include <Rcpp/date_datetime/Datetime.h>  
#include <Rcpp/date_datetime/oldDatetimeVector.h>  
#include <Rcpp/date_datetime/newDatetimeVector.h>
```

Include dependency graph for date\_datetime.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Typedefs

- typedef oldDateVector [Rcpp::DateVector](#)
- typedef oldDatetimeVector [Rcpp::DatetimeVector](#)

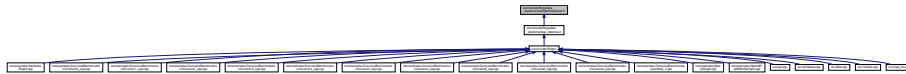
## 7.104 inst/include/Rcpp/date\_datetime/newDatetimeVector.h File Reference

```
#include <RcppCommon.h>
```

Include dependency graph for newDatetimeVector.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::newDatetimeVector](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Functions

- `std::ostream & Rcpp::operator<< (std::ostream &os, const newDatetimeVector d)`

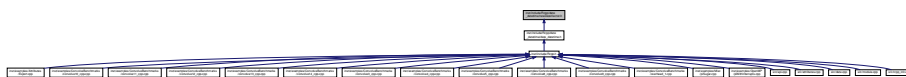
## 7.105 inst/include/Rcpp/date\_datetime/newDateVector.h File Reference

```
#include <RcppCommon.h>
```

Include dependency graph for newDateVector.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::newDateVector](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Functions

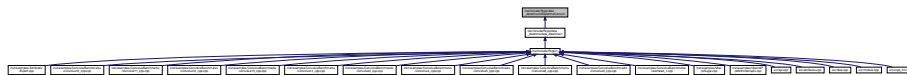
- `std::ostream & Rcpp::operator<< (std::ostream &os, const newDateVector d)`

## 7.106 inst/include/Rcpp/date\_datetime/oldDatetimeVector.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/internal/GreedyVector.h>
Include dependency graph for oldDatetimeVector.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::oldDatetimeVector](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

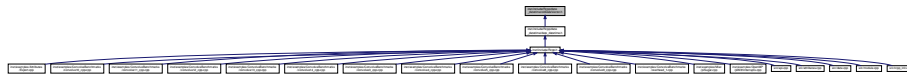
## 7.107 inst/include/Rcpp/date\_datetime/oldDateVector.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/internal/GreedyVector.h>
```

Include dependency graph for oldDateVector.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::oldDateVector](#)

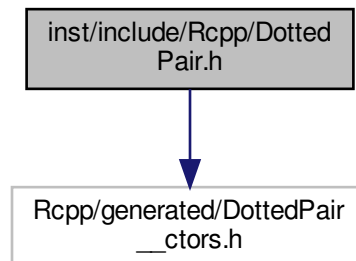
### Namespaces

- [Rcpp](#)  
*Rcpp API.*

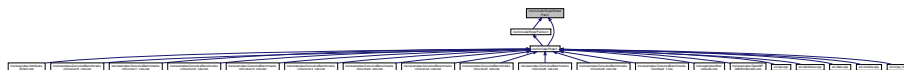
## 7.108 inst/include/Rcpp/DottedPair.h File Reference

```
#include <Rcpp/generated/DottedPair__ctors.h>
```

Include dependency graph for DottedPair.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Typedefs

- typedef DottedPair\_Impl< PreserveStorage > [Rcpp::DottedPair](#)

## Functions

- [Rcpp::RCPP\\_API\\_CLASS](#) (DottedPair\_Impl)

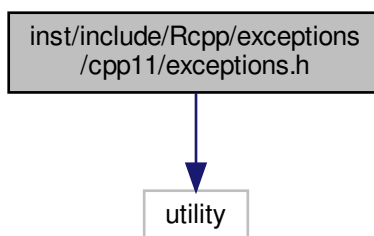
## Variables

- public [Rcpp::DottedPairProxyPolicy](#)< DottedPair\_Impl< StoragePolicy > >
- public public [Rcpp::DottedPairImpl](#)< DottedPair\_Impl< StoragePolicy > >

## 7.109 inst/include/Rcpp/exceptions/cpp11/exceptions.h File Reference

```
#include <utility>
```

Include dependency graph for exceptions.h:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Macros

- #define [RCPP\\_ADVANCED\\_EXCEPTION\\_CLASS](#)(\_\_CLASS\_\_, \_\_WHAT\_\_)

## Functions

- template<typename... Args>  
void [Rcpp::warning](#) (const char \*fmt, Args &&... args)
- template<typename... Args>  
void [NORET Rcpp::stop](#) (const char \*fmt, Args &&... args)

### 7.109.1 Macro Definition Documentation

#### 7.109.1.1 RCPP\_ADVANCED\_EXCEPTION\_CLASS

```
#define RCPP_ADVANCED_EXCEPTION_CLASS(
    __CLASS__,
    __WHAT__ )
```

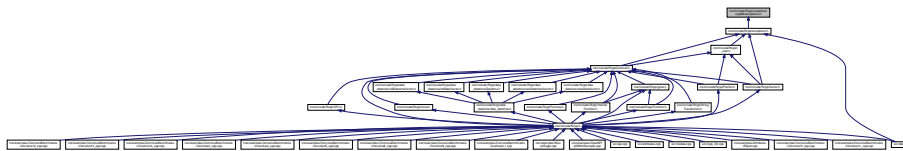
#### Value:

```
class __CLASS__ : public std::exception {
public:
    __CLASS__( ) throw() : message( std::string(__WHAT__) + "." ){}
    __CLASS__( const std::string& message ) throw() :
        message( std::string(__WHAT__) + ": " + message + "." ){}
    template <typename... Args>
    __CLASS__( const char* fmt, Args&&... args ) throw() :
        message( tfm::format(fmt, std::forward<Args>(args)... ) ){}
    virtual ~__CLASS__() throw(){}
    virtual const char* what() const throw() { return message.c_str(); }
private:
    std::string message;
};
```

Definition at line 30 of file exceptions.h.

### 7.110 inst/include/Rcpp/exceptions/cpp98/exceptions.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Macros

- #define [RCPP\\_ADVANCED\\_EXCEPTION\\_CLASS](#)(\_\_CLASS\_\_, \_\_WHAT\_\_)

## Functions

- template<typename T1 >  
void [Rcpp::warning](#) (const char \*fmt, const T1 &arg1)
- template<typename T1, typename T2 >  
void [Rcpp::warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2)
- template<typename T1, typename T2, typename T3 >  
void [Rcpp::warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3)
- template<typename T1, typename T2, typename T3, typename T4 >  
void [Rcpp::warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4)
- template<typename T1, typename T2, typename T3, typename T4, typename T5 >  
void [Rcpp::warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5)
- template<typename T1, typename T2, typename T3, typename T4, typename T5, typename T6 >  
void [Rcpp::warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6)
- template<typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7 >  
void [Rcpp::warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7)
- template<typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7, typename T8 >  
void [Rcpp::warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8)
- template<typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7, typename T8, typename T9 >  
void [Rcpp::warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9)
- template<typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7, typename T8, typename T9, typename T10 >  
void [Rcpp::warning](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9, const T10 &arg10)
- template<typename T1 >  
void [NORET Rcpp::stop](#) (const char \*fmt, const T1 &arg1)
- template<typename T1, typename T2 >  
void [NORET Rcpp::stop](#) (const char \*fmt, const T1 &arg1, const T2 &arg2)
- template<typename T1, typename T2, typename T3 >  
void [NORET Rcpp::stop](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3)
- template<typename T1, typename T2, typename T3, typename T4 >  
void [NORET Rcpp::stop](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4)
- template<typename T1, typename T2, typename T3, typename T4, typename T5 >  
void [NORET Rcpp::stop](#) (const char \*fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5)

- `template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 >`  
`void NORET Rcpp::stop (const char *fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6)`
- `template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 >`  
`void NORET Rcpp::stop (const char *fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7)`
- `template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 >`  
`void NORET Rcpp::stop (const char *fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8)`
- `template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 >`  
`void NORET Rcpp::stop (const char *fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9)`
- `template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename T9 , typename T10 >`  
`void NORET Rcpp::stop (const char *fmt, const T1 &arg1, const T2 &arg2, const T3 &arg3, const T4 &arg4, const T5 &arg5, const T6 &arg6, const T7 &arg7, const T8 &arg8, const T9 &arg9, const T10 &arg10)`

## 7.110.1 Macro Definition Documentation

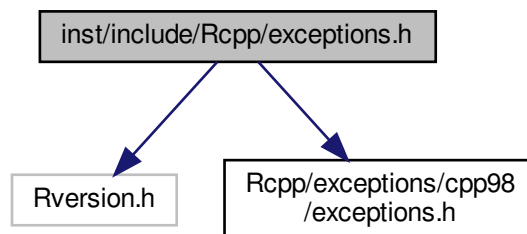
### 7.110.1.1 RCPP\_ADVANCED\_EXCEPTION\_CLASS

```
#define RCPP_ADVANCED_EXCEPTION_CLASS(
    __CLASS__,
    __WHAT__ )
```

Definition at line 28 of file exceptions.h.

## 7.111 inst/include/Rcpp/exceptions.h File Reference

```
#include <Rversion.h>
#include <Rcpp/exceptions/cpp98/exceptions.h>
Include dependency graph for exceptions.h:
```





This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::exception](#)
- class [Rcpp::no\\_such\\_env](#)
- class [Rcpp::file\\_io\\_error](#)
- class [Rcpp::file\\_not\\_found](#)
- class [Rcpp::file\\_exists](#)
- struct [Rcpp::LongjumpException](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Macros

- #define [RCPP\\_DEFAULT\\_INCLUDE\\_CALL](#) true
- #define [GET\\_STACKTRACE\(\)](#) R\_NilValue
- #define [RCPP\\_EXCEPTION\\_CLASS](#)(\_\_CLASS\_\_, \_\_WHAT\_\_)
- #define [RCPP\\_SIMPLE\\_EXCEPTION\\_CLASS](#)(\_\_CLASS\_\_, \_\_MESSAGE\_\_)
- #define [DEMANGLE](#)(\_\_TYPE\_\_) [demangle](#)( typeid(\_\_TYPE\_\_).name() ).c\_str()

## Functions

- static std::string [Rcpp::toString](#) (const int i)
- void [Rcpp::warning](#) (const std::string &message)
- void [NORET Rcpp::stop](#) (const std::string &message)
- SEXP [Rcpp::internal::longjumpSentinel](#) (SEXP token)
- bool [Rcpp::internal::isLongjumpSentinel](#) (SEXP x)
- SEXP [Rcpp::internal::getLongjumpToken](#) (SEXP sentinel)
- void [Rcpp::internal::resumeJump](#) (SEXP token)
- [Rcpp::RCPP\\_SIMPLE\\_EXCEPTION\\_CLASS](#) (not\_a\_matrix, "Not a matrix.") [RCPP\\_SIMPLE\\_EXCEPTION\\_↔](#) CLASS(no\_such\_field
- No such field [Rcpp::RCPP\\_EXCEPTION\\_CLASS](#) (reference\_creation\_error, "Error creating [object](#) of reference class") [RCPP\\_ADVANCED\\_EXCEPTION\\_CLASS](#)(not\_compatible
- SEXP [Rcpp::internal::nth](#) (SEXP s, int n)

- bool [Rcpp::internal::is\\_Rcpp\\_eval\\_call](#) (SEXP expr)
- SEXP [get\\_last\\_call](#) ()
- SEXP [get\\_exception\\_classes](#) (const std::string &ex\_class)
- SEXP [make\\_condition](#) (const std::string &ex\_msg, SEXP call, SEXP cppstack, SEXP classes)
- template<typename Exception >  
SEXP [exception\\_to\\_condition\\_template](#) (const Exception &ex, bool include\_call)
- SEXP [rcpp\\_exception\\_to\\_r\\_condition](#) (const [Rcpp::exception](#) &ex)
- SEXP [exception\\_to\\_r\\_condition](#) (const std::exception &ex)
- SEXP [string\\_to\\_try\\_error](#) (const std::string &str)
- SEXP [exception\\_to\\_try\\_error](#) (const std::exception &ex)
- std::string [demangle](#) (const std::string &name)
- void [forward\\_exception\\_to\\_r](#) (const std::exception &ex)
- void [forward\\_rcpp\\_exception\\_to\\_r](#) (const [Rcpp::exception](#) &ex)

## 7.111.1 Macro Definition Documentation

### 7.111.1.1 DEMANGLE

```
#define DEMANGLE(  
    __TYPE__ ) demangle( typeid(__TYPE__).name() ).c_str()
```

Definition at line 382 of file exceptions.h.

### 7.111.1.2 GET\_STACKTRACE

```
#define GET_STACKTRACE( ) R_NilValue
```

Definition at line 31 of file exceptions.h.

### 7.111.1.3 RCPP\_DEFAULT\_INCLUDE\_CALL

```
#define RCPP_DEFAULT_INCLUDE_CALL true
```

Definition at line 28 of file exceptions.h.

### 7.111.1.4 RCPP\_EXCEPTION\_CLASS

```
#define RCPP_EXCEPTION_CLASS(
    __CLASS__,
    __WHAT__ )
```

**Value:**

```
class __CLASS__ : public std::exception{
public:
    __CLASS__( ) throw() : message( std::string(__WHAT__) + "." ){} ;
    __CLASS__( const std::string& message ) throw() :
    message( std::string(__WHAT__) + ": " + message + "." ){} ;
    virtual ~__CLASS__() throw(){} ;
    virtual const char* what() const throw() { return message.c_str() ; }
private:
    std::string message ;
} ;
```

Definition at line 187 of file exceptions.h.

### 7.111.1.5 RCPP\_SIMPLE\_EXCEPTION\_CLASS

```
#define RCPP_SIMPLE_EXCEPTION_CLASS(
    __CLASS__,
    __MESSAGE__ )
```

**Value:**

```
class __CLASS__ : public std::exception{
public:
    __CLASS__() throw() {} ;
    virtual ~__CLASS__() throw(){} ;
    virtual const char* what() const throw() { return __MESSAGE__ ; }
} ;
```

Definition at line 199 of file exceptions.h.

## 7.111.2 Function Documentation

### 7.111.2.1 demangle()

```
std::string demangle (
    const std::string & name ) [inline]
```

Definition at line 192 of file routines.h.

References GET\_CALLABLE.

Referenced by exception\_to\_condition\_template(), Rcpp::get\_return\_type\_dispatch(), and registerFunctions().

### 7.111.2.2 `exception_to_condition_template()`

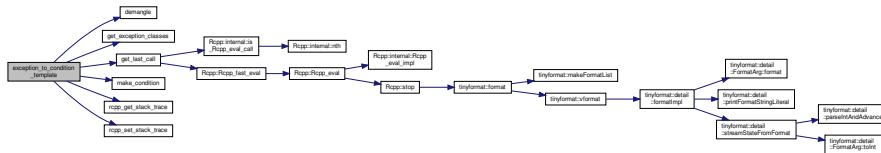
```
template<typename Exception >
SEXP exception_to_condition_template (
    const Exception & ex,
    bool include_call ) [inline]
```

Definition at line 324 of file exceptions.h.

References `demangle()`, `get_exception_classes()`, `get_last_call()`, `make_condition()`, `rcpp_get_stack_trace()`, and `rcpp_set_stack_trace()`.

Referenced by `exception_to_r_condition()`, and `rcpp_exception_to_r_condition()`.

Here is the call graph for this function:



### 7.111.2.3 `exception_to_r_condition()`

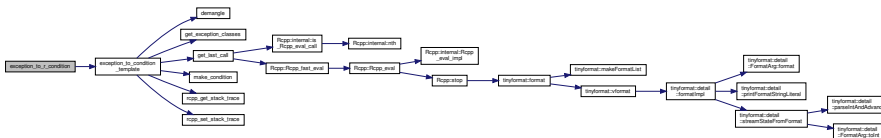
```
SEXP exception_to_r_condition (
    const std::exception & ex ) [inline]
```

Definition at line 352 of file exceptions.h.

References `exception_to_condition_template()`, and `RCPP_DEFAULT_INCLUDE_CALL`.

Referenced by `forward_exception_to_r()`, and `forward_rcpp_exception_to_r()`.

Here is the call graph for this function:



## 7.111.2.4 exception\_to\_try\_error()

```
SEXP exception_to_try_error (
    const std::exception & ex ) [inline]
```

Definition at line 376 of file exceptions.h.

References [string\\_to\\_try\\_error\(\)](#).

Here is the call graph for this function:



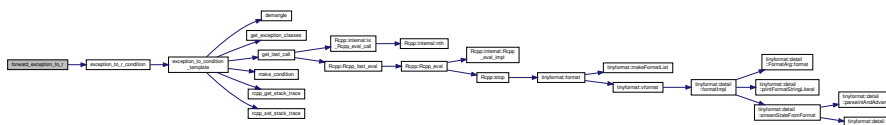
## 7.111.2.5 forward\_exception\_to\_r()

```
void forward_exception_to_r (
    const std::exception & ex ) [inline]
```

Definition at line 386 of file exceptions.h.

References [exception\\_to\\_r\\_condition\(\)](#).

Here is the call graph for this function:

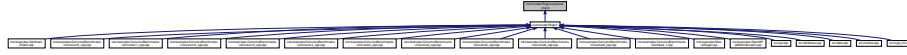






## 7.112 inst/include/Rcpp/exceptions\_impl.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [Rcpp](#)  
*Rcpp API.*

### Macros

- `#define RCPP\_DEMANGLER\_ENABLED 0`

### 7.112.1 Macro Definition Documentation

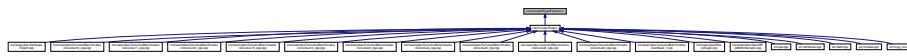
#### 7.112.1.1 RCPP\_DEMANGLER\_ENABLED

```
#define RCPP_DEMANGLER_ENABLED 0
```

Definition at line 41 of file exceptions\_impl.h.

## 7.113 inst/include/Rcpp/Extractor.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::traits::Extractor](#)< [RTYPE](#), [NA](#), [VECTOR](#) >

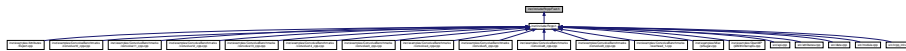


## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.114 inst/include/Rcpp/Fast.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::Fast< VECTOR >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.115 inst/include/Rcpp/Formula.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/Language.h>
```

Include dependency graph for Formula.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Typedefs

- typedef Formula\_Impl< PreserveStorage > [Rcpp::Formula](#)

## Functions

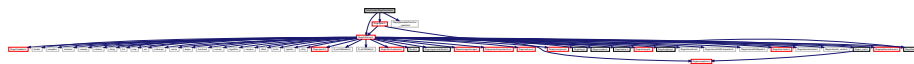
- [Rcpp::RCPP\\_API\\_CLASS](#) (Formula\_Impl)
- [Rcpp::Formula\\_Impl](#) (SEXP x)
- [Rcpp::Formula\\_Impl](#) (const std::string &code)
- void [Rcpp::update](#) (SEXP)

## Variables

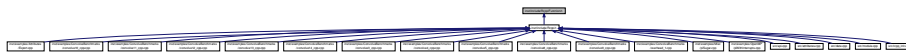
- public [Rcpp::DottedPairProxyPolicy](#)< Formula\_Impl< StoragePolicy > >
- public public [Rcpp::DottedPairImpl](#)< Formula\_Impl< StoragePolicy > >

## 7.116 inst/include/Rcpp/Function.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/grow.h>
#include <Rcpp/generated/Function__operator.h>
Include dependency graph for Function.h:
```



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Typedefs

- typedef Function\_Impl< PreserveStorage > [Rcpp::Function](#)

## Functions

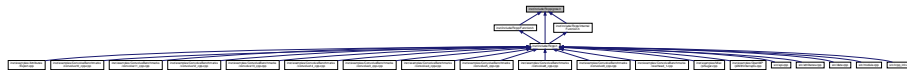
- [Rcpp::RCPP\\_API\\_CLASS](#) (Function\_Impl)

## 7.117 inst/include/Rcpp/grow.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/Named.h>
#include <Rcpp/generated/grow__pairlist.h>
Include dependency graph for grow.h:
```



This graph shows which files directly or indirectly include this file:



## Namespaces

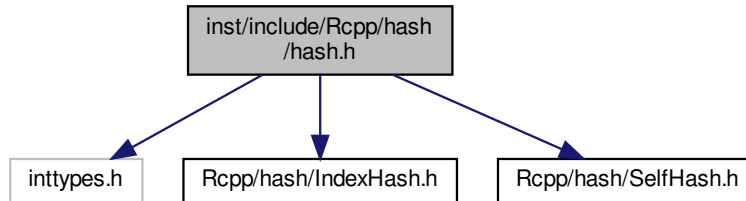
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Functions

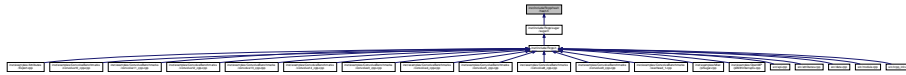
- SEXP [Rcpp::pairlist](#) ()
- SEXP [Rcpp::grow](#) (SEXP head, SEXP tail)
- `template<typename T >`  
SEXP [Rcpp::internal::grow\\_\\_dispatch](#) (`::Rcpp::traits::false_type`, const T &head, SEXP tail)
- `template<typename T >`  
SEXP [Rcpp::internal::grow\\_\\_dispatch](#) (`::Rcpp::traits::true_type`, const T &head, SEXP tail)
- `template<typename T >`  
SEXP [Rcpp::grow](#) (const T &head, SEXP tail)
- SEXP [Rcpp::grow](#) (const char \*head, SEXP tail)

## 7.118 inst/include/Rcpp/hash/hash.h File Reference

```
#include <inttypes.h>
#include <Rcpp/hash/IndexHash.h>
#include <Rcpp/hash/SelfHash.h>
Include dependency graph for hash.h:
```

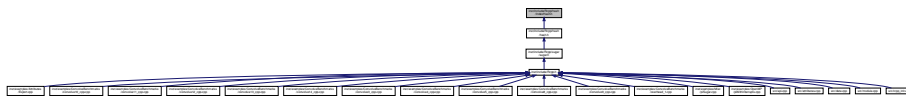


This graph shows which files directly or indirectly include this file:



## 7.119 inst/include/Rcpp/hash/IndexHash.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::IndexHash< RTYPE >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Macros

- `#define RCPP_PROFILE_TIC`
- `#define RCPP_PROFILE_TOC`
- `#define RCPP_PROFILE_RECORD(name)`
- `#define RCPP_USE_CACHE_HASH`
- `#define RCPP_HASH(X) (3141592653U * ((uint32_t)(X)) >> (32 - k))`

### 7.119.1 Macro Definition Documentation

#### 7.119.1.1 RCPP\_HASH

```
#define RCPP_HASH(  
    X ) (3141592653U * ((uint32_t)(X)) >> (32 - k))
```

Definition at line 45 of file IndexHash.h.

#### 7.119.1.2 RCPP\_PROFILE\_RECORD

```
#define RCPP_PROFILE_RECORD(  
    name )
```

Definition at line 37 of file IndexHash.h.

#### 7.119.1.3 RCPP\_PROFILE\_TIC

```
#define RCPP_PROFILE_TIC
```

Definition at line 35 of file IndexHash.h.

#### 7.119.1.4 RCPP\_PROFILE\_TOC

```
#define RCPP_PROFILE_TOC
```

Definition at line 36 of file IndexHash.h.

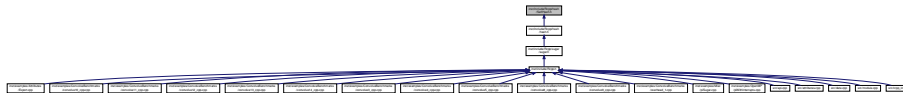
### 7.119.1.5 RCPP\_USE\_CACHE\_HASH

```
#define RCPP_USE_CACHE_HASH
```

Definition at line 39 of file IndexHash.h.

## 7.120 inst/include/Rcpp/hash/SelfHash.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::SelfHash< RTYPE >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## 7.121 inst/include/Rcpp/InputParameter.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

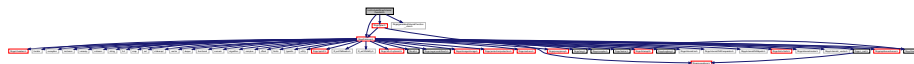
- class [Rcpp::InputParameter< T >](#)
- class [Rcpp::ReferenceInputParameter< T >](#)
- class [Rcpp::ConstInputParameter< T >](#)
- class [Rcpp::ConstReferenceInputParameter< T >](#)
- struct [Rcpp::traits::input\\_parameter< T >](#)
- struct [Rcpp::traits::input\\_parameter< T & >](#)
- struct [Rcpp::traits::input\\_parameter< const T >](#)
- struct [Rcpp::traits::input\\_parameter< const T & >](#)

## Namespaces

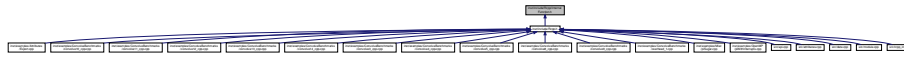
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.122 inst/include/Rcpp/InternalFunction.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/grow.h>
#include <Rcpp/generated/InternalFunction__ctors.h>
Include dependency graph for InternalFunction.h:
```



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Typedefs

- typedef InternalFunction\_Impl< PreserveStorage > [Rcpp::InternalFunction](#)

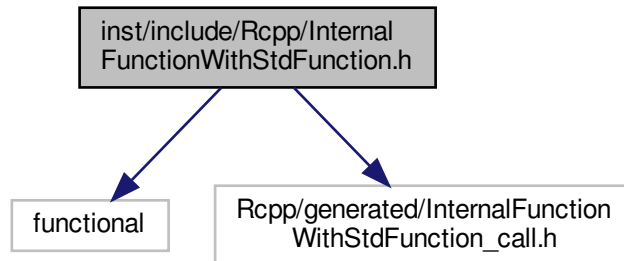
## Functions

- [Rcpp::RCPP\\_API\\_CLASS](#) (InternalFunction\_Impl)

## 7.123 inst/include/Rcpp/InternalFunctionWithStdFunction.h File Reference

```
#include <functional>
#include <Rcpp/generated/InternalFunctionWithStdFunction_call.h>
```

Include dependency graph for InternalFunctionWithStdFunction.h:



### Classes

- class [Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< RESULT\\_TYPE, Args >](#)
- class [Rcpp::InternalFunctionWithStdFunction::CppFunctionBaseFromStdFunction< void, Args... >](#)

### Namespaces

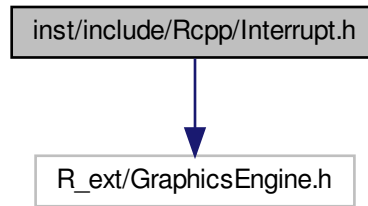
- [Rcpp](#)
  - *Rcpp API.*
- [Rcpp::InternalFunctionWithStdFunction](#)

## 7.124 inst/include/Rcpp/Interrupt.h File Reference

```
#include <R_ext/GraphicsEngine.h>
```



Include dependency graph for Interrupt.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::internal::InterruptedException](#)

## Namespaces

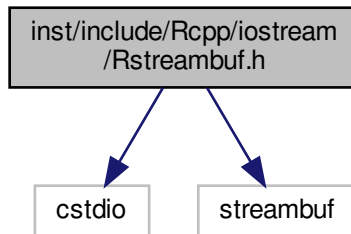
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Functions

- SEXP [Rcpp::internal::interruptedError](#) ()
- void [Rcpp::checkUserInterrupt](#) ()

## 7.125 inst/include/Rcpp/iostream/Rstreambuf.h File Reference

```
#include <cstdio>
#include <streambuf>
Include dependency graph for Rstreambuf.h:
```



This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::Rstreambuf< OUTPUT >](#)
- class [Rcpp::Rostream< OUTPUT >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*

### Variables

- static [Rostream< true >](#) [Rcpp::Rcout](#) = [Rcpp\\_cout\\_get\(\)](#)
- static [Rostream< false >](#) [Rcpp::Rcerr](#) = [Rcpp\\_cerr\\_get\(\)](#)

## 7.126 inst/include/Rcpp/lang.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [Rcpp](#)  
*Rcpp API.*

### Macros

- `#define Rcpp\_list1 Rf_list1`
- `#define Rcpp\_lang1 Rf_lang1`
- `#define Rcpp\_lang2 Rf_lang2`
- `#define Rcpp\_lang3 Rf_lang3`
- `#define Rcpp\_lang4 Rf_lang4`
- `#define Rcpp\_lang5 Rf_lang5`
- `#define Rcpp\_lang6 Rf_lang6`
- `#define Rcpp\_icons Rf_icons`

### Functions

- `SEXP Rcpp::Rcpp\_list2 (SEXP x0, SEXP x1)`
- `SEXP Rcpp::Rcpp\_list3 (SEXP x0, SEXP x1, SEXP x2)`
- `SEXP Rcpp::Rcpp\_list4 (SEXP x0, SEXP x1, SEXP x2, SEXP x3)`
- `SEXP Rcpp::Rcpp\_list5 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4)`
- `SEXP Rcpp::Rcpp\_list6 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5)`
- `SEXP Rcpp::Rcpp\_list7 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6)`
- `SEXP Rcpp::Rcpp\_lang7 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6)`
- `SEXP Rcpp::Rcpp\_list8 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7)`
- `SEXP Rcpp::Rcpp\_lang8 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7)`
- `SEXP Rcpp::Rcpp\_list9 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8)`
- `SEXP Rcpp::Rcpp\_lang9 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8)`
- `SEXP Rcpp::Rcpp\_list10 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9)`
- `SEXP Rcpp::Rcpp\_lang10 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9)`
- `SEXP Rcpp::Rcpp\_list11 (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10)`

- SEXP [Rcpp::Rcpp\\_lang11](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10)
- SEXP [Rcpp::Rcpp\\_list12](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11)
- SEXP [Rcpp::Rcpp\\_lang12](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11)
- SEXP [Rcpp::Rcpp\\_list13](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12)
- SEXP [Rcpp::Rcpp\\_lang13](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12)
- SEXP [Rcpp::Rcpp\\_list14](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13)
- SEXP [Rcpp::Rcpp\\_lang14](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13)
- SEXP [Rcpp::Rcpp\\_list15](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14)
- SEXP [Rcpp::Rcpp\\_lang15](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14)
- SEXP [Rcpp::Rcpp\\_list16](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15)
- SEXP [Rcpp::Rcpp\\_lang16](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15)
- SEXP [Rcpp::Rcpp\\_list17](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16)
- SEXP [Rcpp::Rcpp\\_lang17](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16)
- SEXP [Rcpp::Rcpp\\_list18](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16, SEXP x17)
- SEXP [Rcpp::Rcpp\\_lang18](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16, SEXP x17)
- SEXP [Rcpp::Rcpp\\_list19](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16, SEXP x17, SEXP x18)
- SEXP [Rcpp::Rcpp\\_lang19](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16, SEXP x17, SEXP x18)
- SEXP [Rcpp::Rcpp\\_list20](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16, SEXP x17, SEXP x18, SEXP x19)
- SEXP [Rcpp::Rcpp\\_lang20](#) (SEXP x0, SEXP x1, SEXP x2, SEXP x3, SEXP x4, SEXP x5, SEXP x6, SEXP x7, SEXP x8, SEXP x9, SEXP x10, SEXP x11, SEXP x12, SEXP x13, SEXP x14, SEXP x15, SEXP x16, SEXP x17, SEXP x18, SEXP x19)

## 7.126.1 Macro Definition Documentation

### 7.126.1.1 Rcpp\_lang1

```
#define Rcpp_lang1 Rf_lang1
```

Definition at line 25 of file lang.h.

### 7.126.1.2 Rcpp\_lang2

```
#define Rcpp_lang2 Rf_lang2
```

Definition at line 26 of file lang.h.

### 7.126.1.3 Rcpp\_lang3

```
#define Rcpp_lang3 Rf_lang3
```

Definition at line 27 of file lang.h.

### 7.126.1.4 Rcpp\_lang4

```
#define Rcpp_lang4 Rf_lang4
```

Definition at line 28 of file lang.h.

### 7.126.1.5 Rcpp\_lang5

```
#define Rcpp_lang5 Rf_lang5
```

Definition at line 29 of file lang.h.

### 7.126.1.6 Rcpp\_lang6

```
#define Rcpp_lang6 Rf_lang6
```

Definition at line 30 of file lang.h.

### 7.126.1.7 Rcpp\_lcons

```
#define Rcpp_lcons Rf_lcons
```

Definition at line 32 of file lang.h.

### 7.126.1.8 Rcpp\_list1

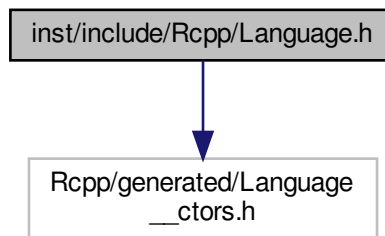
```
#define Rcpp_list1 Rf_list1
```

Definition at line 24 of file lang.h.

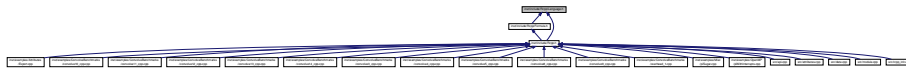
## 7.127 inst/include/Rcpp/Language.h File Reference

```
#include <Rcpp/generated/Language__ctors.h>
```

Include dependency graph for Language.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::fixed\\_call< RESULT\\_TYPE >](#)
- class [Rcpp::unary\\_call< T, RESULT\\_TYPE >](#)
- class [Rcpp::binary\\_call< T1, T2, RESULT\\_TYPE >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Typedefs

- typedef DottedPairProxyPolicy< Language\_Impl >::const\_DottedPairProxy [Rcpp::const\\_Proxy](#)
- typedef Language\_Impl< PreserveStorage > [Rcpp::Language](#)

## Functions

- [Rcpp::RCPP\\_API\\_CLASS](#) (Language\_Impl)
- [Rcpp::Language\\_Impl](#) ()
- [Rcpp::Language\\_Impl](#) (SEXP x)
- [Rcpp::Language\\_Impl](#) (const std::string &symbol)
- [Rcpp::Language\\_Impl](#) (const Symbol &symbol)
- [Rcpp::Language\\_Impl](#) (const Function &function)
- void [Rcpp::setSymbol](#) (const std::string &symbol)
- void [Rcpp::setSymbol](#) (const Symbol &symbol)
- void [Rcpp::setFunction](#) (const Function &function)
- SEXP [Rcpp::eval](#) () const
- SEXP [Rcpp::eval](#) (SEXP env) const
- SEXP [Rcpp::fast\\_eval](#) () const
- SEXP [Rcpp::fast\\_eval](#) (SEXP env) const
- void [Rcpp::update](#) (SEXP)

## Variables

- public [Rcpp::DottedPairProxyPolicy](#)< Language\_Impl< StoragePolicy > >
- public public [Rcpp::DottedPairImpl](#)< Language\_Impl< StoragePolicy > >

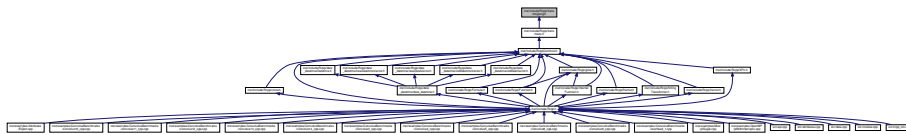
## 7.128 inst/include/Rcpp/longlong.h File Reference

This graph shows which files directly or indirectly include this file:



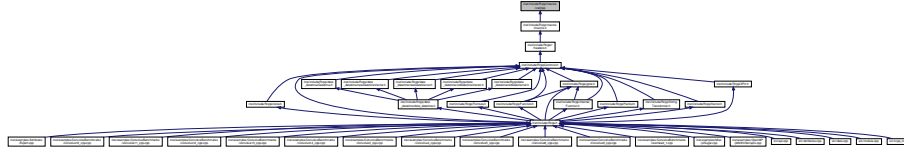
## 7.129 inst/include/Rcpp/traits/longlong.h File Reference

This graph shows which files directly or indirectly include this file:



## 7.130 inst/include/Rcpp/macros/cat.hpp File Reference

This graph shows which files directly or indirectly include this file:



### Macros

- `#define RCPP_PP_CAT(a, b) RCPP_PP_CAT_OO((a, b))`
- `#define RCPP_PP_CAT_OO(par) RCPP_PP_CAT_I ## par`
- `#define RCPP_PP_CAT_I(a, b) RCPP_PP_CAT_II(a ## b)`
- `#define RCPP_PP_CAT_II(res) res`

### 7.130.1 Macro Definition Documentation

#### 7.130.1.1 RCPP\_PP\_CAT

```
#define RCPP_PP_CAT(  
    a,  
    b ) RCPP_PP_CAT_OO((a, b))
```

Definition at line 22 of file cat.hpp.

#### 7.130.1.2 RCPP\_PP\_CAT\_I

```
#define RCPP_PP_CAT_I(  
    a,  
    b ) RCPP_PP_CAT_II(a ## b)
```

Definition at line 29 of file cat.hpp.



### 7.130.1.3 RCPP\_PP\_CAT\_II

```
#define RCPP_PP_CAT_II(  
    res ) res
```

Definition at line 30 of file cat.hpp.

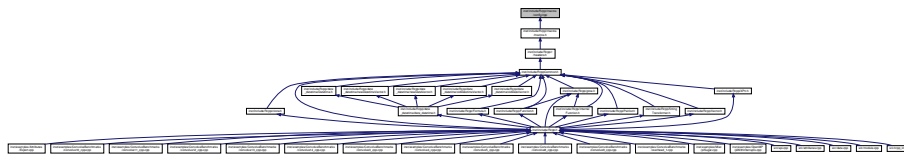
### 7.130.1.4 RCPP\_PP\_CAT\_OO

```
#define RCPP_PP_CAT_OO(  
    par ) RCPP_PP_CAT_I ## par
```

Definition at line 23 of file cat.hpp.

## 7.131 inst/include/Rcpp/macros/config.hpp File Reference

This graph shows which files directly or indirectly include this file:



## Macros

- #define [RCPP\\_PP\\_CONFIG\\_STRICT\(\)](#) 0x0001
- #define [RCPP\\_PP\\_CONFIG\\_IDEAL\(\)](#) 0x0002
- #define [RCPP\\_PP\\_CONFIG\\_MSVC\(\)](#) 0x0004
- #define [RCPP\\_PP\\_CONFIG\\_MWCC\(\)](#) 0x0008
- #define [RCPP\\_PP\\_CONFIG\\_BCC\(\)](#) 0x0010
- #define [RCPP\\_PP\\_CONFIG\\_EDG\(\)](#) 0x0020
- #define [RCPP\\_PP\\_CONFIG\\_DMC\(\)](#) 0x0040
- #define [RCPP\\_PP\\_CONFIG\\_FLAGS\(\)](#) ([RCPP\\_PP\\_CONFIG\\_STRICT\(\)](#))
- #define [RCPP\\_PP\\_CONFIG\\_EXTENDED\\_LINE\\_INFO](#) 0
- #define [RCPP\\_PP\\_CONFIG\\_ERRORS](#) 1

### 7.131.1 Macro Definition Documentation

#### 7.131.1.1 RCPP\_PP\_CONFIG\_BCC

```
#define RCPP_PP_CONFIG_BCC( ) 0x0010
```

Definition at line 22 of file config.hpp.

#### 7.131.1.2 RCPP\_PP\_CONFIG\_DMC

```
#define RCPP_PP_CONFIG_DMC( ) 0x0040
```

Definition at line 24 of file config.hpp.

#### 7.131.1.3 RCPP\_PP\_CONFIG\_EDG

```
#define RCPP_PP_CONFIG_EDG( ) 0x0020
```

Definition at line 23 of file config.hpp.

#### 7.131.1.4 RCPP\_PP\_CONFIG\_ERRORS

```
#define RCPP_PP_CONFIG_ERRORS 1
```

Definition at line 66 of file config.hpp.

#### 7.131.1.5 RCPP\_PP\_CONFIG\_EXTENDED\_LINE\_INFO

```
#define RCPP_PP_CONFIG_EXTENDED_LINE_INFO 0
```

Definition at line 57 of file config.hpp.

#### 7.131.1.6 RCPP\_PP\_CONFIG\_FLAGS

```
#define RCPP_PP_CONFIG_FLAGS( ) (RCPP_PP_CONFIG_STRICT())
```

Definition at line 50 of file config.hpp.

### 7.131.1.7 RCPP\_PP\_CONFIG\_IDEAL

```
#define RCPP_PP_CONFIG_IDEAL( ) 0x0002
```

Definition at line 18 of file config.hpp.

### 7.131.1.8 RCPP\_PP\_CONFIG\_MSVC

```
#define RCPP_PP_CONFIG_MSVC( ) 0x0004
```

Definition at line 20 of file config.hpp.

### 7.131.1.9 RCPP\_PP\_CONFIG\_MWCC

```
#define RCPP_PP_CONFIG_MWCC( ) 0x0008
```

Definition at line 21 of file config.hpp.

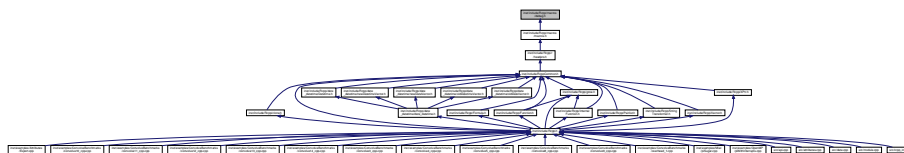
### 7.131.1.10 RCPP\_PP\_CONFIG\_STRICT

```
#define RCPP_PP_CONFIG_STRICT( ) 0x0001
```

Definition at line 17 of file config.hpp.

## 7.132 inst/include/Rcpp/macros/debug.h File Reference

This graph shows which files directly or indirectly include this file:



## Macros

- #define [RCPP\\_DEBUG\\_LEVEL](#) 0
- #define [RCPP\\_DEBUG\\_MODULE\\_LEVEL](#) [RCPP\\_DEBUG\\_LEVEL](#)
- #define [RCPP\\_DEBUG](#)(MSG)
- #define [RCPP\\_DEBUG\\_1](#)(fmt, MSG)
- #define [RCPP\\_DEBUG\\_2](#)(fmt, M1, M2)
- #define [RCPP\\_DEBUG\\_3](#)(fmt, M1, M2, M3)
- #define [RCPP\\_DEBUG\\_4](#)(fmt, M1, M2, M3, M4)
- #define [RCPP\\_DEBUG\\_5](#)(fmt, M1, M2, M3, M4, M5)
- #define [RCPP\\_DEBUG\\_MODULE](#)(MSG)
- #define [RCPP\\_DEBUG\\_MODULE\\_1](#)(fmt, MSG)
- #define [RCPP\\_DEBUG\\_MODULE\\_2](#)(fmt, M1, M2)
- #define [RCPP\\_DEBUG\\_MODULE\\_3](#)(fmt, M1, M2, M3)
- #define [RCPP\\_DEBUG\\_MODULE\\_4](#)(fmt, M1, M2, M3, M4)
- #define [RCPP\\_DEBUG\\_MODULE\\_5](#)(fmt, M1, M2, M3, M4, M5)

### 7.132.1 Macro Definition Documentation

#### 7.132.1.1 RCPP\_DEBUG

```
#define RCPP_DEBUG(  
    MSG )
```

Definition at line 43 of file debug.h.

#### 7.132.1.2 RCPP\_DEBUG\_1

```
#define RCPP_DEBUG_1(  
    fmt,  
    MSG )
```

Definition at line 44 of file debug.h.

#### 7.132.1.3 RCPP\_DEBUG\_2

```
#define RCPP_DEBUG_2(  
    fmt,  
    M1,  
    M2 )
```

Definition at line 45 of file debug.h.

#### 7.132.1.4 RCPP\_DEBUG\_3

```
#define RCPP_DEBUG_3(  
    fmt,  
    M1,  
    M2,  
    M3 )
```

Definition at line 46 of file debug.h.

#### 7.132.1.5 RCPP\_DEBUG\_4

```
#define RCPP_DEBUG_4(  
    fmt,  
    M1,  
    M2,  
    M3,  
    M4 )
```

Definition at line 47 of file debug.h.

#### 7.132.1.6 RCPP\_DEBUG\_5

```
#define RCPP_DEBUG_5(  
    fmt,  
    M1,  
    M2,  
    M3,  
    M4,  
    M5 )
```

Definition at line 48 of file debug.h.

#### 7.132.1.7 RCPP\_DEBUG\_LEVEL

```
#define RCPP_DEBUG_LEVEL 0
```

Definition at line 27 of file debug.h.

### 7.132.1.8 RCPP\_DEBUG\_MODULE

```
#define RCPP_DEBUG_MODULE(  
    MSG )
```

Definition at line 101 of file debug.h.

### 7.132.1.9 RCPP\_DEBUG\_MODULE\_1

```
#define RCPP_DEBUG_MODULE_1(  
    fmt,  
    MSG )
```

Definition at line 102 of file debug.h.

### 7.132.1.10 RCPP\_DEBUG\_MODULE\_2

```
#define RCPP_DEBUG_MODULE_2(  
    fmt,  
    M1,  
    M2 )
```

Definition at line 103 of file debug.h.

### 7.132.1.11 RCPP\_DEBUG\_MODULE\_3

```
#define RCPP_DEBUG_MODULE_3(  
    fmt,  
    M1,  
    M2,  
    M3 )
```

Definition at line 104 of file debug.h.

#### 7.132.1.12 RCPP\_DEBUG\_MODULE\_4

```
#define RCPP_DEBUG_MODULE_4(  
    fmt,  
    M1,  
    M2,  
    M3,  
    M4 )
```

Definition at line 105 of file debug.h.

#### 7.132.1.13 RCPP\_DEBUG\_MODULE\_5

```
#define RCPP_DEBUG_MODULE_5(  
    fmt,  
    M1,  
    M2,  
    M3,  
    M4,  
    M5 )
```

Definition at line 106 of file debug.h.

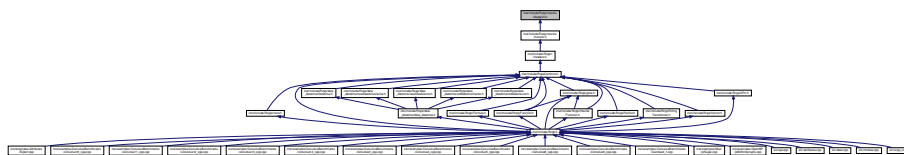
#### 7.132.1.14 RCPP\_DEBUG\_MODULE\_LEVEL

```
#define RCPP_DEBUG_MODULE_LEVEL RCPP\_DEBUG\_LEVEL
```

Definition at line 31 of file debug.h.

## 7.133 inst/include/Rcpp/macros/dispatch.h File Reference

This graph shows which files directly or indirectly include this file:



## Macros

- #define `__RCPP_HANDLE_CASE__`( `__RTYPE__`, `__FUN__`, `__OBJECT__`, `__RCPPTYPE__` )
- #define `__RCPP_RETURN__`( `__FUN__`, `__SEXP__`, `__RCPPTYPE__` )
- #define `RCPP_RETURN_VECTOR`( `__FUN__`, `__SEXP__` ) `__RCPP_RETURN__`( `__FUN__`, `__SEXP__`, `Vector` )
- #define `RCPP_RETURN_MATRIX`( `__FUN__`, `__SEXP__` ) `__RCPP_RETURN__`( `__FUN__`, `__SEXP__`, `Matrix` )

### 7.133.1 Macro Definition Documentation

#### 7.133.1.1 `__RCPP_HANDLE_CASE__`

```
#define __RCPP_HANDLE_CASE__(
    __RTYPE__,
    __FUN__,
    __OBJECT__,
    __RCPPTYPE__ )
```

#### Value:

```
case __RTYPE__:
    return __FUN__(: :Rcpp::__RCPPTYPE__ < __RTYPE__ > (__OBJECT__));
```

Definition at line 82 of file dispatch.h.

#### 7.133.1.2 `__RCPP_RETURN__`

```
#define __RCPP_RETURN__(
    __FUN__,
    __SEXP__,
    __RCPPTYPE__ )
```

#### Value:

```
SEXP __TMP__ = __SEXP__;
switch (TYPEOF(__TMP__)) {
    __RCPP_HANDLE_CASE__(INTSXP, __FUN__, __TMP__, __RCPPTYPE__)
    __RCPP_HANDLE_CASE__(REALSXP, __FUN__, __TMP__, __RCPPTYPE__)
    __RCPP_HANDLE_CASE__(RAWSXP, __FUN__, __TMP__, __RCPPTYPE__)
    __RCPP_HANDLE_CASE__(LGLSXP, __FUN__, __TMP__, __RCPPTYPE__)
    __RCPP_HANDLE_CASE__(CPLXSXP, __FUN__, __TMP__, __RCPPTYPE__)
    __RCPP_HANDLE_CASE__(STRSXP, __FUN__, __TMP__, __RCPPTYPE__)
    __RCPP_HANDLE_CASE__(VECSXP, __FUN__, __TMP__, __RCPPTYPE__)
    __RCPP_HANDLE_CASE__(EXPRSXP, __FUN__, __TMP__, __RCPPTYPE__)
default:
    throw std::range_error("Not a vector");
}
```

Definition at line 87 of file dispatch.h.



### 7.133.1.3 RCPP\_RETURN\_MATRIX

```
#define RCPP_RETURN_MATRIX(  
    _FUN_,  
    _SEXP_ )  \_\_\_\_RCPP\_RETURN\_\_\_\_(_FUN_, _SEXP_, Matrix)
```

Definition at line 104 of file dispatch.h.

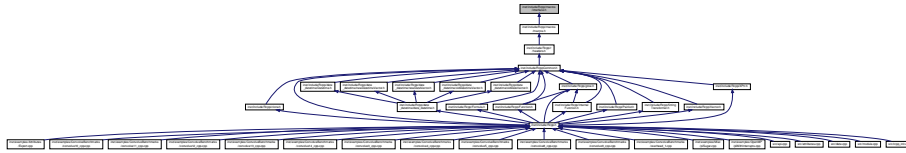
### 7.133.1.4 RCPP\_RETURN\_VECTOR

```
#define RCPP_RETURN_VECTOR(  
    _FUN_,  
    _SEXP_ )  \_\_\_\_RCPP\_RETURN\_\_\_\_(_FUN_, _SEXP_, Vector)
```

Definition at line 102 of file dispatch.h.

## 7.134 inst/include/Rcpp/macros/interface.h File Reference

This graph shows which files directly or indirectly include this file:



## Macros

- #define [RCPP\\_GENERATE\\_CTOR\\_ASSIGN](#)(\_\_CLASS\_\_)
- #define [RCPP\\_CTOR\\_ASSIGN](#)(\_\_CLASS\_\_)
- #define [RCPP\\_CTOR\\_ASSIGN\\_WITH\\_BASE](#)(\_\_CLASS\_\_)
- #define [RCPP\\_API\\_CLASS](#)(\_\_CLASS\_\_)

### 7.134.1 Macro Definition Documentation

### 7.134.1.1 RCPP\_API\_CLASS

```
#define RCPP_API_CLASS(  
    __CLASS__ )
```

**Value:**

```
template < template <class> class StoragePolicy > class __CLASS__ :  
    public StoragePolicy<__CLASS__<StoragePolicy> >,  
    public SlotProxyPolicy<__CLASS__<StoragePolicy> >,  
    public AttributeProxyPolicy<__CLASS__<StoragePolicy> >,  
    public ROjectMethods< __CLASS__<StoragePolicy> >
```

Definition at line 49 of file interface.h.

### 7.134.1.2 RCPP\_CTOR\_ASSIGN

```
#define RCPP_CTOR_ASSIGN(  
    __CLASS__ )
```

**Value:**

```
__CLASS__( const __CLASS__& other ){  
    Storage::copy__(other) ;  
}  
__CLASS__& operator=(const __CLASS__& rhs) {  
    return Storage::copy__(rhs) ;  
}  
template <typename Proxy>  
__CLASS__( const GenericProxy<Proxy>& proxy ){  
    Storage::set__( proxy.get() ) ;  
}
```

Definition at line 26 of file interface.h.

### 7.134.1.3 RCPP\_CTOR\_ASSIGN\_WITH\_BASE

```
#define RCPP_CTOR_ASSIGN_WITH_BASE(  
    __CLASS__ )
```

**Value:**

```
__CLASS__( const __CLASS__& other ) : Base(other) {  
}  
__CLASS__& operator=(const __CLASS__& rhs) {  
    return Storage::copy__(rhs) ;  
}  
template <typename Proxy>  
__CLASS__( const GenericProxy<Proxy>& proxy ){  
    Storage::set__( proxy.get() ) ;  
}
```

Definition at line 38 of file interface.h.

## 7.134.1.4 RCPP\_GENERATE\_CTOR\_ASSIGN

```
#define RCPP_GENERATE_CTOR_ASSIGN(
    __CLASS__ )
```

**Value:**

```
typedef StoragePolicy<__CLASS__> Storage ;
typedef AttributeProxyPolicy<__CLASS__> AttributePolicy ;
RCPP_CTOR_ASSIGN(__CLASS__)
```

Definition at line 21 of file interface.h.

## 7.135 inst/include/Rcpp/macros/macros.h File Reference

```
#include <string>
#include <Rcpp/macros/debug.h>
#include <Rcpp/macros/unroll.h>
#include <Rcpp/macros/dispatch.h>
#include <Rcpp/macros/xp.h>
#include <Rcpp/macros/traits.h>
#include <Rcpp/macros/config.hpp>
#include <Rcpp/macros/cat.hpp>
#include <Rcpp/macros/module.h>
#include <Rcpp/macros/interface.h>
```

Include dependency graph for macros.h:



This graph shows which files directly or indirectly include this file:

**Namespaces**

- [Rcpp](#)
  - Rcpp API.*
- [Rcpp::internal](#)
  - internal implementation details*
- [Rcpp::internal::debug](#)

## Macros

- #define [RCPP\\_DECORATE](#)(\_\_FUN\_\_) \_\_FUN\_\_##\_rcpp\_wrapper\_\_
- #define [RCPP\\_GET\\_NAMES](#)(x) Rf\_getAttrib(x, R\_NamesSymbol)
- #define [RCPP\\_GET\\_CLASS](#)(x) Rf\_getAttrib(x, R\_ClassSymbol)
- #define [BEGIN\\_RCPP](#)
- #define [VOID\\_END\\_RCPP](#)
- #define [END\\_RCPP VOID\\_END\\_RCPP](#) return R\_NilValue;
- #define [END\\_RCPP\\_RETURN\\_ERROR](#)
- #define [Rcpp\\_error](#)(MESSAGE) throw [Rcpp::exception](#)(MESSAGE, \_\_FILE\_\_, \_\_LINE\_\_)

## Functions

- std::string [Rcpp::internal::debug::short\\_file\\_name](#) (const char \*file)

### 7.135.1 Macro Definition Documentation

#### 7.135.1.1 BEGIN\_RCPP

```
#define BEGIN_RCPP
```

##### Value:

```
int rcpp_output_type = 0 ;
int nprot = 0;
(void)rcpp_output_type;
SEXP rcpp_output_condition = R_NilValue ;
(void)rcpp_output_condition;
static SEXP stop_sym = Rf_install("stop");
try {
```

```
//
//
//
//
//
```

Definition at line 49 of file macros.h.

#### 7.135.1.2 END\_RCPP

```
#define END_RCPP VOID_END_RCPP return R_NilValue;
```

Definition at line 99 of file macros.h.

### 7.135.1.3 END\_RCPP\_RETURN\_ERROR

```
#define END_RCPP_RETURN_ERROR
```

**Value:**

```

}
catch (Rcpp::internal::InterruptedException &__ex__) {
    return Rcpp::internal::interruptedError();
}
catch (Rcpp::LongjumpException& __ex__) {
    return Rcpp::internal::longjumpSentinel(__ex__.token);
}
catch (std::exception &__ex__) {
    return exception_to_try_error(__ex__);
}
catch (...) {
    return string_to_try_error("c++ exception (unknown reason)");
}
UNPROTECT(nprot);
return R_NilValue;
(void) stop_sym; /* never reached but suppresses warning */

```

Definition at line 106 of file macros.h.

### 7.135.1.4 RCPP\_DECORATE

```
#define RCPP_DECORATE(
    __FUN__ ) __FUN__##_rcpp_wrapper__
```

Definition at line 44 of file macros.h.

### 7.135.1.5 Rcpp\_error

```
#define Rcpp_error(
    MESSAGE ) throw Rcpp::exception(MESSAGE, __FILE__, __LINE__)
```

Definition at line 125 of file macros.h.

### 7.135.1.6 RCPP\_GET\_CLASS

```
#define RCPP_GET_CLASS(
    x ) Rf_getAttrib(x, R_ClassSymbol)
```

Definition at line 46 of file macros.h.

### 7.135.1.7 RCPP\_GET\_NAMES

```
#define RCPP_GET_NAMES(
    x ) Rf_getAttrib(x, R_NamesSymbol)
```

Definition at line 45 of file macros.h.

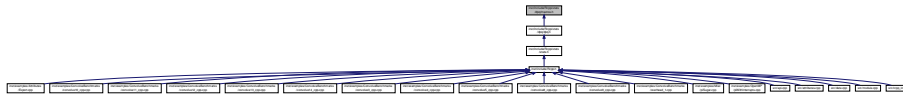
### 7.135.1.8 VOID\_END\_RCPP

```
#define VOID_END_RCPP
```

Definition at line 60 of file macros.h.

## 7.136 inst/include/Rcpp/stats/dpq/macros.h File Reference

This graph shows which files directly or indirectly include this file:



### Macros

- #define [give\\_log](#) log\_p
- #define [R\\_D\\_\\_0](#) (log\_p ? [ML\\_NEGINF](#) : 0.) /\* 0 \*/
- #define [R\\_D\\_\\_1](#) (log\_p ? 0. : 1.) /\* 1 \*/
- #define [R\\_DT\\_0](#) (lower\_tail ? [R\\_D\\_\\_0](#) : [R\\_D\\_\\_1](#)) /\* 0 \*/
- #define [R\\_DT\\_1](#) (lower\_tail ? [R\\_D\\_\\_1](#) : [R\\_D\\_\\_0](#)) /\* 1 \*/
- #define [R\\_D\\_Lval](#)(p) (lower\_tail ? (p) : (0.5 - (p) + 0.5)) /\* p \*/
- #define [R\\_D\\_Cval](#)(p) (lower\_tail ? (0.5 - (p) + 0.5) : (p)) /\* 1 - p \*/
- #define [R\\_D\\_val](#)(x) (log\_p ? ::log(x) : (x)) /\* x in pF(x,..) \*/
- #define [R\\_D\\_qlv](#)(p) (log\_p ? ::exp(p) : (p)) /\* p in qF(p,..) \*/
- #define [R\\_D\\_exp](#)(x) (log\_p ? (x) : ::exp(x)) /\* exp(x) \*/
- #define [R\\_D\\_log](#)(p) (log\_p ? (p) : ::log(p)) /\* log(p) \*/
- #define [R\\_D\\_Clog](#)(p) (log\_p ? ::log1p(-(p)) : (0.5 - (p) + 0.5)) /\* [log](1-p) \*/
- #define [R\\_Log1\\_Exp](#)(x) ((x) > -M\_LN2 ? ::log(-::expm1(x)) : ::log1p(-::exp(x)))
- #define [R\\_D\\_LExp](#)(x) (log\_p ? [R\\_Log1\\_Exp](#)(x) : ::log1p(-x))
- #define [R\\_DT\\_val](#)(x) (lower\_tail ? [R\\_D\\_val](#)(x) : [R\\_D\\_Clog](#)(x))
- #define [R\\_DT\\_Cval](#)(x) (lower\_tail ? [R\\_D\\_Clog](#)(x) : [R\\_D\\_val](#)(x))
- #define [R\\_DT\\_qlv](#)(p)
- #define [R\\_DT\\_Clv](#)(p)
- #define [R\\_DT\\_exp](#)(x) [R\\_D\\_exp](#)([R\\_D\\_Lval](#)(x)) /\* exp(x) \*/
- #define [R\\_DT\\_Cexp](#)(x) [R\\_D\\_exp](#)([R\\_D\\_Cval](#)(x)) /\* exp(1 - x) \*/

- `#define R_DT_log(p) (lower_tail? R_D_log(p) : R_D_LExp(p))/* log(p) in qF */`
- `#define R_DT_Clog(p) (lower_tail? R_D_LExp(p): R_D_log(p))/* log(1-p) in qF*/`
- `#define R_DT_Log(p) (lower_tail? (p) : R_Log1_Exp(p))`
- `#define R_Q_P01_check(p)`
- `#define R_Q_P01_boundaries(p, _LEFT_, _RIGHT_)`
- `#define R_P_bounds_01(x, x_min, x_max)`
- `#define R_P_bounds_Inf_01(x)`
- `#define R_D_fexp(f, x) (give_log ? -0.5*log(f)+(x) : exp(x)/sqrt(f))`
- `#define R_D_forceint(x) floor((x) + 0.5)`
- `#define R_D_nonint(x) (fabs((x) - floor((x)+0.5)) > 1e-7)`
- `#define R_D_neglnonint(x) (x < 0. || R_D_nonint(x))`
- `#define R_D_nonint_check(x)`

## 7.136.1 Macro Definition Documentation

### 7.136.1.1 give\_log

```
#define give_log log_p
```

Definition at line 24 of file macros.h.

### 7.136.1.2 R\_D\_\_0

```
#define R_D__0 (log_p ? ML_NEGINF : 0.) /* 0 */
```

Definition at line 27 of file macros.h.

### 7.136.1.3 R\_D\_\_1

```
#define R_D__1 (log_p ? 0. : 1.) /* 1 */
```

Definition at line 28 of file macros.h.

#### 7.136.1.4 R\_D\_Clog

```
#define R_D_Clog(  
    p ) (log_p ? ::log1p(-(p)) : (0.5 - (p) + 0.5)) /* [log](1-p) */
```

Definition at line 40 of file macros.h.

#### 7.136.1.5 R\_D\_Cval

```
#define R_D_Cval(  
    p ) (lower_tail ? (0.5 - (p) + 0.5) : (p)) /* 1 - p */
```

Definition at line 34 of file macros.h.

#### 7.136.1.6 R\_D\_exp

```
#define R_D_exp(  
    x ) (log_p ? (x) : ::exp(x)) /* exp(x) */
```

Definition at line 38 of file macros.h.

#### 7.136.1.7 R\_D\_fexp

```
#define R_D_fexp(  
    f,  
    x ) (give_log ? -0.5*log(f)+(x) : exp(x)/sqrt(f))
```

Definition at line 116 of file macros.h.

#### 7.136.1.8 R\_D\_forceint

```
#define R_D_forceint(  
    x ) floor((x) + 0.5)
```

Definition at line 117 of file macros.h.



### 7.136.1.9 R\_D\_LExp

```
#define R_D_LExp(  
    x ) (log_p ? R_Log1_Exp(x) : ::log1p(-x))
```

Definition at line 46 of file macros.h.

### 7.136.1.10 R\_D\_log

```
#define R_D_log(  
    p ) (log_p ? (p) : ::log(p)) /* log(p) */
```

Definition at line 39 of file macros.h.

### 7.136.1.11 R\_D\_Lval

```
#define R_D_Lval(  
    p ) (lower_tail ? (p) : (0.5 - (p) + 0.5)) /* p */
```

Definition at line 33 of file macros.h.

### 7.136.1.12 R\_D\_negInonint

```
#define R_D_negInonint(  
    x ) (x < 0. || R_D_nonint(x))
```

Definition at line 120 of file macros.h.

### 7.136.1.13 R\_D\_nonint

```
#define R_D_nonint(  
    x ) (fabs((x) - floor((x)+0.5)) > 1e-7)
```

Definition at line 118 of file macros.h.

#### 7.136.1.14 R\_D\_nonint\_check

```
#define R_D_nonint_check(  
    x )
```

**Value:**

```
if(R_D_nonint(x)) {  
    MATHLIB_WARNING("non-integer x = %f", x);  
    return R_D__0;  
}
```

Definition at line 122 of file macros.h.

#### 7.136.1.15 R\_D\_qIv

```
#define R_D_qIv(  
    p ) (log_p ? ::exp(p) : (p)) /* p in qF(p,..) */
```

Definition at line 37 of file macros.h.

#### 7.136.1.16 R\_D\_val

```
#define R_D_val(  
    x ) (log_p ? ::log(x) : (x)) /* x in pF(x,..) */
```

Definition at line 36 of file macros.h.

#### 7.136.1.17 R\_DT\_0

```
#define R_DT_0 (lower_tail ? R_D__0 : R_D__1) /* 0 */
```

Definition at line 29 of file macros.h.

#### 7.136.1.18 R\_DT\_1

```
#define R_DT_1 (lower_tail ? R_D__1 : R_D__0) /* 1 */
```

Definition at line 30 of file macros.h.

### 7.136.1.19 R\_DT\_Cexp

```
#define R_DT_Cexp(  
    x ) R_D_exp(R_D_Cval(x)) /* exp(1 - x) */
```

Definition at line 60 of file macros.h.

### 7.136.1.20 R\_DT\_Clv

```
#define R_DT_Clv(  
    p )
```

**Value:**

```
(log_p ? (lower_tail ? -expm1(p) : ::exp(p)) \  
: R_D_Cval(p))
```

Definition at line 56 of file macros.h.

### 7.136.1.21 R\_DT\_Clog

```
#define R_DT_Clog(  
    p ) (lower_tail? R_D_LExp(p): R_D_log(p))/* log(1-p) in qF*/
```

Definition at line 63 of file macros.h.

### 7.136.1.22 R\_DT\_Cval

```
#define R_DT_Cval(  
    x ) (lower_tail ? R_D_Clog(x) : R_D_val(x))
```

Definition at line 49 of file macros.h.

### 7.136.1.23 R\_DT\_exp

```
#define R_DT_exp(  
    x ) R_D_exp(R_D_Lval(x)) /* exp(x) */
```

Definition at line 59 of file macros.h.

**7.136.1.24 R\_DT\_log**

```
#define R_DT_log(
    p ) (lower_tail? R_D_log(p) : R_D_LExp(p))/* log(p) in qF */
```

Definition at line 62 of file macros.h.

**7.136.1.25 R\_DT\_Log**

```
#define R_DT_Log(
    p ) (lower_tail? (p) : R_Log1_Exp(p))
```

Definition at line 64 of file macros.h.

**7.136.1.26 R\_DT\_qIv**

```
#define R_DT_qIv(
    p )
```

**Value:**

```
(log_p ? (lower_tail ? ::exp(p) : - ::expm1(p)) \
: R_D_Lval(p))
```

Definition at line 52 of file macros.h.

**7.136.1.27 R\_DT\_val**

```
#define R_DT_val(
    x ) (lower_tail ? R_D_val(x) : R_D_Clog(x))
```

Definition at line 48 of file macros.h.

**7.136.1.28 R\_Log1\_Exp**

```
#define R_Log1_Exp(
    x ) ((x) > -M_LN2 ? ::log(-::expm1(x)) : ::log1p(-::exp(x)))
```

Definition at line 43 of file macros.h.

**7.136.1.29 R\_P\_bounds\_01**

```
#define R_P_bounds_01(
    x,
    x_min,
    x_max )
```

**Value:**

```
if(x <= x_min) return R_DT_0;
if(x >= x_max) return R_DT_1
```

Definition at line 102 of file macros.h.

**7.136.1.30 R\_P\_bounds\_Inf\_01**

```
#define R_P_bounds_Inf_01(
    x )
```

**Value:**

```
if(!R_FINITE(x)) {
    if (x > 0) return R_DT_1;
    /* x < 0 */return R_DT_0;
}
```

Definition at line 107 of file macros.h.

**7.136.1.31 R\_Q\_P01\_boundaries**

```
#define R_Q_P01_boundaries(
    p,
    _LEFT_,
    _RIGHT_ )
```

**Value:**

```
if (log_p) {
    if(p > 0)
        return R_NaN ;
    if(p == 0) /* upper bound*/
        return lower_tail ? _RIGHT_ : _LEFT_;
    if(p == ML_NEGINF)
        return lower_tail ? _LEFT_ : _RIGHT_;
}
else { /* !log_p */
    if(p < 0 || p > 1)
        return R_NaN ;
    if(p == 0)
        return lower_tail ? _LEFT_ : _RIGHT_;
    if(p == 1)
        return lower_tail ? _RIGHT_ : _LEFT_;
}
```

Definition at line 84 of file macros.h.

### 7.136.1.32 R\_Q\_P01\_check

```
#define R_Q_P01_check(
    p )
```

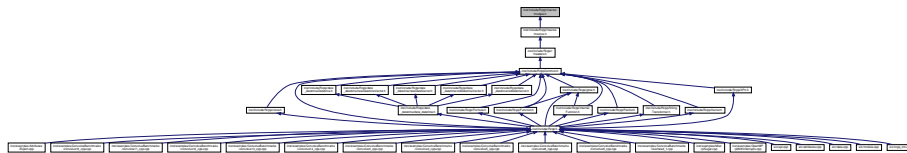
#### Value:

```
if ((log_p && p > 0) ||
    (!log_p && (p < 0 || p > 1)) ) \
    return R_NaN
```

Definition at line 68 of file macros.h.

## 7.137 inst/include/Rcpp/macros/module.h File Reference

This graph shows which files directly or indirectly include this file:



## Macros

- #define [RCPP\\_EXPOSED\\_AS\(CLASS\)](#)
- #define [RCPP\\_EXPOSED\\_WRAP\(CLASS\)](#) namespace Rcpp{ namespace traits{ template<> struct wrap\_← type\_traits< CLASS >{typedef wrap\_type\_module\_object\_tag wrap\_category ; } ;}}
- #define [RCPP\\_EXPOSED\\_CLASS\\_NODECL\(CLASS\)](#)
- #define [RCPP\\_EXPOSED\\_CLASS\(CLASS\)](#)
- #define [RCPP\\_EXPOSED\\_ENUM\\_AS\(CLASS\)](#) namespace Rcpp{ namespace traits{ template<> struct r\_← type\_traits< CLASS >{ typedef r\_type\_enum\_tag r\_category ; } ;}}
- #define [RCPP\\_EXPOSED\\_ENUM\\_WRAP\(CLASS\)](#) namespace Rcpp{ namespace traits{ template<> struct wrap\_type\_traits< CLASS >{typedef wrap\_type\_enum\_tag wrap\_category ; } ;}}
- #define [RCPP\\_EXPOSED\\_ENUM\\_NODECL\(CLASS\)](#)
- #define [RCPP\\_EXPOSED\\_ENUM\(CLASS\)](#)

### 7.137.1 Macro Definition Documentation

#### 7.137.1.1 RCPP\_EXPOSED\_AS

```
#define RCPP_EXPOSED_AS(
    CLASS )
```

This macros should be used by packages using modules when a type is used as a parameter of a function or method exposed by modules. This defines the necessary trait that makes the class as<>'able

Definition at line 29 of file module.h.

### 7.137.1.2 RCPP\_EXPOSED\_CLASS

```
#define RCPP_EXPOSED_CLASS(  
    CLASS )
```

**Value:**

```
class CLASS; \  
RCPP_EXPOSED_CLASS_NODECL(CLASS)
```

Definition at line 69 of file module.h.

### 7.137.1.3 RCPP\_EXPOSED\_CLASS\_NODECL

```
#define RCPP_EXPOSED_CLASS_NODECL(  
    CLASS )
```

**Value:**

```
RCPP_EXPOSED_AS(CLASS) \  
RCPP_EXPOSED_WRAP(CLASS)
```

Definition at line 65 of file module.h.

### 7.137.1.4 RCPP\_EXPOSED\_ENUM

```
#define RCPP_EXPOSED_ENUM(  
    CLASS )
```

**Value:**

```
class CLASS; \  
RCPP_EXPOSED_ENUM_NODECL(CLASS)
```

Definition at line 83 of file module.h.

### 7.137.1.5 RCPP\_EXPOSED\_ENUM\_AS

```
#define RCPP_EXPOSED_ENUM_AS(  
    CLASS ) namespace Rcpp{ namespace traits{ template<> struct r_type_traits< CLASS  
>{ typedef r_type_enum_tag r_category ; } ; }}
```

handling enums: TODO use `is_enum` from C++11 or boost to have those automatic

Definition at line 76 of file module.h.

### 7.137.1.6 RCPP\_EXPOSED\_ENUM\_NODECL

```
#define RCPP_EXPOSED_ENUM_NODECL(  
    CLASS )
```

#### Value:

```
RCPP_EXPOSED_ENUM_AS(CLASS) \  
RCPP_EXPOSED_ENUM_WRAP(CLASS)
```

Definition at line 79 of file module.h.

### 7.137.1.7 RCPP\_EXPOSED\_ENUM\_WRAP

```
#define RCPP_EXPOSED_ENUM_WRAP(  
    CLASS ) namespace Rcpp{ namespace traits{ template<> struct wrap_type_traits< CLASS  
>{typedef wrap_type_enum_tag wrap_category ; } ; }}
```

Definition at line 77 of file module.h.

### 7.137.1.8 RCPP\_EXPOSED\_WRAP

```
#define RCPP_EXPOSED_WRAP(  
    CLASS ) namespace Rcpp{ namespace traits{ template<> struct wrap_type_traits< CLASS  
>{typedef wrap_type_module_object_tag wrap_category ; } ; }}
```

Definition at line 63 of file module.h.

## 7.138 inst/include/Rcpp/macros/traits.h File Reference

This graph shows which files directly or indirectly include this file:



### Macros

- #define [RCPP\\_TRAITS](#)(\_\_CLASS\_\_, \_\_SEXPTYPE\_\_)
- #define [RCPP\\_ENUM\\_TRAITS](#)(\_\_ENUM\_\_) [RCPP\\_TRAITS](#)(\_\_ENUM\_\_,INTSXP)



## 7.138.1 Macro Definition Documentation

### 7.138.1.1 RCPP\_ENUM\_TRAITS

```
#define RCPP_ENUM_TRAITS(
    __ENUM__ ) RCPP_TRAITS(__ENUM__, INTSXP)
```

Definition at line 40 of file traits.h.

### 7.138.1.2 RCPP\_TRAITS

```
#define RCPP_TRAITS(
    __CLASS__,
    __SEXPTYPE__ )
```

#### Value:

```
namespace Rcpp{ namespace traits {
template<> struct r_type_traits< __CLASS__ >{
    typedef r_type_primitive_tag r_category ;
};
template<> struct r_type_traits< std::pair< std::string , __CLASS__ > >{
    typedef r_type_pairstring_primitive_tag r_category ;
};
template<> struct wrap_type_traits< __CLASS__ >{
    typedef wrap_type_primitive_tag wrap_category ;
};
template<> struct r_sexptype_traits< __CLASS__ >{
    enum{ rtype = __SEXPTYPE__ } ;
};
} }
```

Definition at line 25 of file traits.h.

## 7.139 inst/include/Rcpp/traits/traits.h File Reference

```
#include <Rcpp/traits/integral_constant.h>
#include <Rcpp/traits/same_type.h>
#include <Rcpp/traits/enable_if.h>
#include <Rcpp/traits/is_wide_string.h>
#include <Rcpp/traits/is_arithmetic.h>
#include <Rcpp/traits/char_type.h>
#include <Rcpp/traits/named_object.h>
#include <Rcpp/traits/is_convertible.h>
#include <Rcpp/traits/has_iterator.h>
#include <Rcpp/traits/expands_to_logical.h>
#include <Rcpp/traits/matrix_interface.h>
#include <Rcpp/traits/is_sugar_expression.h>
#include <Rcpp/traits/is_eigen_base.h>
```

```

#include <Rcpp/traits/has_na.h>
#include <Rcpp/traits/storage_type.h>
#include <Rcpp/traits/r_sexptype_traits.h>
#include <Rcpp/traits/r_type_traits.h>
#include <Rcpp/traits/un_pointer.h>
#include <Rcpp/traits/is_pointer.h>
#include <Rcpp/traits/wrap_type_traits.h>
#include <Rcpp/traits/longlong.h>
#include <Rcpp/traits/module_wrap_traits.h>
#include <Rcpp/traits/is_na.h>
#include <Rcpp/traits/is_finite.h>
#include <Rcpp/traits/is_infinite.h>
#include <Rcpp/traits/is_nan.h>
#include <Rcpp/traits/is_bool.h>
#include <Rcpp/traits/if_.h>
#include <Rcpp/traits/get_na.h>
#include <Rcpp/traits/is_trivial.h>
#include <Rcpp/traits/init_type.h>
#include <Rcpp/traits/is_const.h>
#include <Rcpp/traits/is_reference.h>
#include <Rcpp/traits/remove_const.h>
#include <Rcpp/traits/remove_reference.h>
#include <Rcpp/traits/remove_const_and_reference.h>
#include <Rcpp/traits/result_of.h>
#include <Rcpp/traits/is_module_object.h>
#include <Rcpp/traits/is_primitive.h>
#include <Rcpp/traits/one_type.h>

```

Include dependency graph for traits.h:



This graph shows which files directly or indirectly include this file:



## Classes

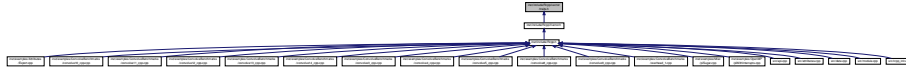
- struct [Rcpp::traits::identity< T >](#)
- struct [Rcpp::traits::int2type< I >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.140 inst/include/Rcpp/vector/traits.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

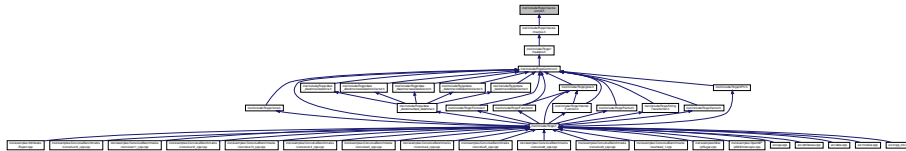
- class [Rcpp::traits::r\\_vector\\_cache< RTYPE, StoragePolicy >](#)
- class [Rcpp::traits::proxy\\_cache< RTYPE, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_cache\\_type< RTYPE, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_cache\\_type< VECSXP, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_cache\\_type< EXPRSXP, StoragePolicy >](#)
- struct [Rcpp::traits::r\\_vector\\_cache\\_type< STRSXP, StoragePolicy >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.141 inst/include/Rcpp/macros/unroll.h File Reference

This graph shows which files directly or indirectly include this file:



### Macros

- #define [RCPP\\_LOOP\\_UNROLL\\_PTR](#)(TARGET, SOURCE)
- #define [RCPP\\_LOOP\\_UNROLL](#)(TARGET, SOURCE)
- #define [RCPP\\_LOOP\\_UNROLL\\_LHSFUN](#)(TARGET, FUN, SOURCE)

### 7.141.1 Macro Definition Documentation

### 7.141.1.1 RCPP\_LOOP\_UNROLL

```
#define RCPP_LOOP_UNROLL(  
    TARGET,  
    SOURCE )
```

#### Value:

```
R_xlen_t __trip_count = n » 2 ;  
R_xlen_t i = 0 ;  
for ( ; __trip_count > 0 ; --__trip_count) {  
    TARGET[i] = SOURCE[i] ; i++ ;  
    TARGET[i] = SOURCE[i] ; i++ ;  
    TARGET[i] = SOURCE[i] ; i++ ;  
    TARGET[i] = SOURCE[i] ; i++ ;  
}  
switch (n - i){  
    case 3:  
        TARGET[i] = SOURCE[i] ; i++ ; /* fallthrough */  
    case 2:  
        TARGET[i] = SOURCE[i] ; i++ ; /* fallthrough */  
    case 1:  
        TARGET[i] = SOURCE[i] ; i++ ; /* fallthrough */  
    case 0:  
    default:  
        {}  
}
```

Definition at line 47 of file unroll.h.

### 7.141.1.2 RCPP\_LOOP\_UNROLL\_LHSFUN

```
#define RCPP_LOOP_UNROLL_LHSFUN(  
    TARGET,  
    FUN,  
    SOURCE )
```

#### Value:

```
R_xlen_t __trip_count = n » 2 ;  
R_xlen_t i = 0 ;  
for ( ; __trip_count > 0 ; --__trip_count) {  
    TARGET[FUN(i)] = SOURCE[i] ; i++ ;  
    TARGET[FUN(i)] = SOURCE[i] ; i++ ;  
    TARGET[FUN(i)] = SOURCE[i] ; i++ ;  
    TARGET[FUN(i)] = SOURCE[i] ; i++ ;  
}  
switch (n - i){  
    case 3:  
        TARGET[FUN(i)] = SOURCE[i] ; i++ ; /* fallthrough */  
    case 2:  
        TARGET[FUN(i)] = SOURCE[i] ; i++ ; /* fallthrough */  
    case 1:  
        TARGET[FUN(i)] = SOURCE[i] ; i++ ; /* fallthrough */  
    case 0:  
    default:  
        {}  
}
```

Definition at line 68 of file unroll.h.

## 7.141.1.3 RCPP\_LOOP\_UNROLL\_PTR

```
#define RCPP_LOOP_UNROLL_PTR(
    TARGET,
    SOURCE )
```

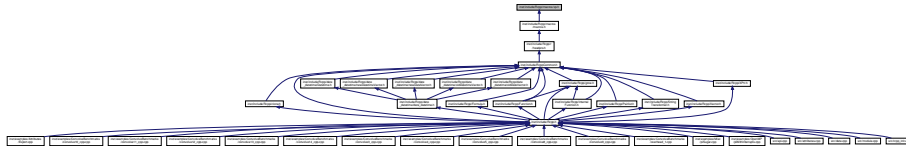
**Value:**

```
R_xlen_t __trip_count = n >> 2 ;
R_xlen_t i = 0 ;
for ( ; __trip_count > 0 ; --__trip_count) {
    *TARGET++ = SOURCE[i++] ;
    *TARGET++ = SOURCE[i++] ;
    *TARGET++ = SOURCE[i++] ;
    *TARGET++ = SOURCE[i++] ;
}
switch (n - i){
  case 3:
    *TARGET++ = SOURCE[i++] ;
  case 2:
    *TARGET++ = SOURCE[i++] ;
  case 1:
    *TARGET++ = SOURCE[i++] ;
  case 0:
  default:
    {}
}
```

Definition at line 25 of file unroll.h.

## 7.142 inst/include/Rcpp/macros/xp.h File Reference

This graph shows which files directly or indirectly include this file:

**Macros**

- #define [RCPP\\_XP\\_FIELD\\_GET](#)(\_\_NAME\_\_, \_\_CLASS\_\_, \_\_FIELD\_\_)
- #define [RCPP\\_XP\\_FIELD\\_SET](#)(\_\_NAME\_\_, \_\_CLASS\_\_, \_\_FIELD\_\_)
- #define [RCPP\\_XP\\_FIELD](#)(\_\_PREFIX\_\_, \_\_CLASS\_\_, \_\_FIELD\_\_)

## 7.142.1 Macro Definition Documentation

### 7.142.1.1 RCPP\_XP\_FIELD

```
#define RCPP_XP_FIELD(
    __PREFIX__,
    __CLASS__,
    __FIELD__ )
```

#### Value:

```
RCPP_XP_FIELD_GET( RCPP_PP_CAT(__PREFIX__,_get), __CLASS__, __FIELD__ ) \
RCPP_XP_FIELD_SET( RCPP_PP_CAT(__PREFIX__,_set), __CLASS__, __FIELD__ )
```

Definition at line 61 of file xp.h.

### 7.142.1.2 RCPP\_XP\_FIELD\_GET

```
#define RCPP_XP_FIELD_GET(
    __NAME__,
    __CLASS__,
    __FIELD__ )
```

#### Value:

```
extern "C" SEXP RCPP_PP_CAT(__NAME__,__rcpp_info__){
    using Rcpp::_;
    Rcpp::List info = Rcpp::List::create(
        _["class"] = #__CLASS__ ,
        _["field"] = #__FIELD__
    ) ;
    info.attr( "class" ) = "rcppxpfieldgetinfo" ;
    return info ;
}
extern "C" SEXP __NAME__( SEXP xp ){
    BEGIN_RCPP
        SEXP res = R_NilValue ;
        ::Rcpp::XPtr< __CLASS__ > ptr(xp) ;
        res = ::Rcpp::wrap( ptr->__FIELD__ ) ;
        return res ;
    END_RCPP
}
```

Definition at line 25 of file xp.h.

### 7.142.1.3 RCPP\_XP\_FIELD\_SET

```
#define RCPP_XP_FIELD_SET(
    __NAME__,
    __CLASS__,
    __FIELD__ )
```

#### Value:

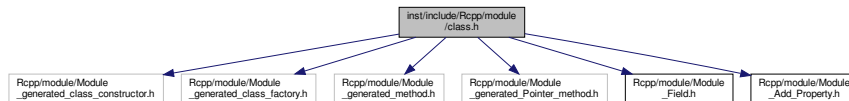
```
extern "C" SEXP RCPP_PP_CAT(__NAME__,__rcpp_info__){
    using Rcpp::_;
    Rcpp::List info = Rcpp::List::create(
        _["class"] = #__CLASS__ ,
        _["field"] = #__FIELD__
    ) ;
    info.attr( "class" ) = "rcppxpfieldsetinfo" ;
    return info ;
}
extern "C" SEXP __NAME__( SEXP xp, SEXP value ){
    BEGIN_RCPP
        ::Rcpp::XPtr< __CLASS__ > ptr(xp) ;
        ptr->__FIELD__ = ::Rcpp::internal::converter(value) ;
    END_RCPP
}
```

Definition at line 44 of file xp.h.

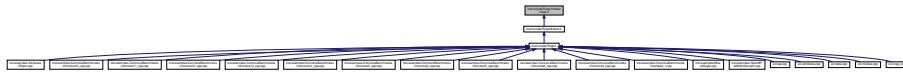
## 7.143 inst/include/Rcpp/module/class.h File Reference

```
#include <Rcpp/module/Module_generated_class_constructor.h>
#include <Rcpp/module/Module_generated_class_factory.h>
#include <Rcpp/module/Module_generated_method.h>
#include <Rcpp/module/Module_generated_Pointer_method.h>
#include <Rcpp/module/Module_Field.h>
#include <Rcpp/module/Module_Add_Property.h>
```

Include dependency graph for class.h:



This graph shows which files directly or indirectly include this file:

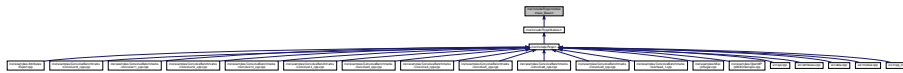


### Classes

- class [class\\_< Class >](#)

## 7.144 inst/include/Rcpp/module/class\_Base.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

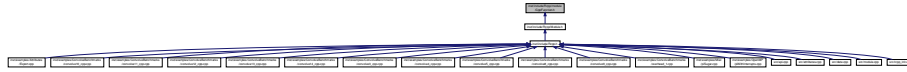
- class [Rcpp::class\\_Base](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.145 inst/include/Rcpp/module/CppFunction.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

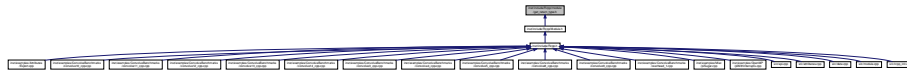
- class [Rcpp::CppFunctionBase](#)
- class [Rcpp::CppFunction](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.146 inst/include/Rcpp/module/get\_return\_type.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::void\\_type](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*

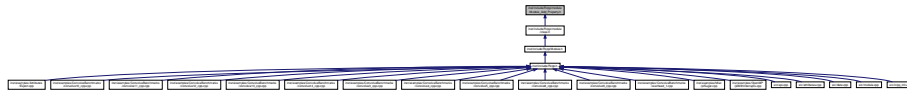


## Functions

- `template<typename RESULT_TYPE >`  
`std::string Rcpp::get_return_type_dispatch (Rcpp::traits::false_type)`
- `template<typename RESULT_TYPE >`  
`std::string Rcpp::get_return_type_dispatch (Rcpp::traits::true_type)`
- `template<typename RESULT_TYPE >`  
`std::string Rcpp::get_return_type ()`
- `template<> std::string Rcpp::get_return_type< void_type > ()`
- `template<> std::string Rcpp::get_return_type< SEXP > ()`
- `template<> std::string Rcpp::get_return_type< Rcpp::IntegerVector > ()`
- `template<> std::string Rcpp::get_return_type< Rcpp::NumericVector > ()`
- `template<> std::string Rcpp::get_return_type< Rcpp::RawVector > ()`
- `template<> std::string Rcpp::get_return_type< Rcpp::ExpressionVector > ()`
- `template<> std::string Rcpp::get_return_type< Rcpp::List > ()`
- `template<> std::string Rcpp::get_return_type< Rcpp::CharacterVector > ()`

## 7.147 inst/include/Rcpp/module/Module\_Add\_Property.h File Reference

This graph shows which files directly or indirectly include this file:



## Functions

- `template<typename PROP >`  
`self & property (const char *name_, PROP(Class::*GetMethod)(void), const char *docstring=0)`
- `template<typename PROP >`  
`self & property (const char *name_, PROP(*GetMethod)(Class *), const char *docstring)`
- `template<typename PROP >`  
`self & property (const char *name_, PROP(Class::*GetMethod)(void), void(Class::*SetMethod)(PROP), const char *docstring=0)`
- `template<typename PROP >`  
`self & property (const char *name_, PROP(Class::*GetMethod)(void), void(*SetMethod)(Class *, PROP), const char *docstring=0)`
- `template<typename PROP >`  
`self & property (const char *name_, PROP(*GetMethod)(Class *), void(Class::*SetMethod)(PROP), const char *docstring=0)`
- `template<typename PROP >`  
`self & property (const char *name_, PROP(*GetMethod)(Class *), void(*SetMethod)(Class *, PROP), const char *docstring=0)`

### 7.147.1 Function Documentation

### 7.147.1.1 `property()` [1/6]

```
template<typename PROP >
self& property (
    const char * name_,
    PROP(*) (Class *) GetMethod,
    const char * docstring )
```

Definition at line 38 of file `Module_Add_Property.h`.

### 7.147.1.2 `property()` [2/6]

```
template<typename PROP >
self& property (
    const char * name_,
    PROP(*) (Class *) GetMethod,
    void(*) (Class *, PROP) SetMethod,
    const char * docstring = 0 )
```

Definition at line 89 of file `Module_Add_Property.h`.

### 7.147.1.3 `property()` [3/6]

```
template<typename PROP >
self& property (
    const char * name_,
    PROP(*) (Class *) GetMethod,
    void(Class::*)(PROP) SetMethod,
    const char * docstring = 0 )
```

Definition at line 81 of file `Module_Add_Property.h`.

### 7.147.1.4 `property()` [4/6]

```
template<typename PROP >
self & property (
    const char * name_,
    PROP(Class::*)(void) GetMethod,
    const char * docstring = 0 )
```

Definition at line 26 of file `Module_Add_Property.h`.

**7.147.1.5 property()** [5/6]

```
template<typename PROP >
self & property (
    const char * name_,
    PROP(Class::*) (void) GetMethod,
    void(*) (Class *, PROP) SetMethod,
    const char * docstring = 0 )
```

Definition at line 63 of file Module\_Add\_Property.h.

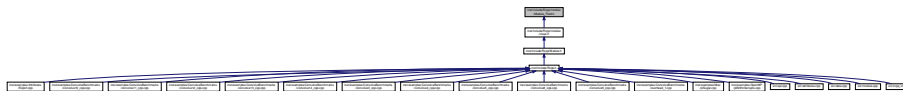
**7.147.1.6 property()** [6/6]

```
template<typename PROP >
self & property (
    const char * name_,
    PROP(Class::*) (void) GetMethod,
    void(Class::*) (PROP) SetMethod,
    const char * docstring = 0 )
```

Definition at line 45 of file Module\_Add\_Property.h.

**7.148 inst/include/Rcpp/module/Module\_Field.h File Reference**

This graph shows which files directly or indirectly include this file:

**Classes**

- class [CppProperty\\_Getter\\_Setter< PROP >](#)
- class [CppProperty\\_Getter< PROP >](#)

**Functions**

- `template<typename T >`  
self & [field](#) (const char \*name\_, T Class::\*ptr, const char \*docstring=0)
- `template<typename T >`  
self & [field\\_readonly](#) (const char \*name\_, T Class::\*ptr, const char \*docstring=0)

## 7.148.1 Function Documentation

### 7.148.1.1 field()

```
template<typename T >
self& field (
    const char * name_,
    T Class::* ptr,
    const char * docstring = 0 )
```

Definition at line 68 of file Module\_Field.h.

Referenced by Rcpp::S4\_CppConstructor< Class >::S4\_CppConstructor(), Rcpp::S4\_CppOverloadedMethods< Class >::S4\_CppOverloadedMethods(), and Rcpp::S4\_field< Class >::S4\_field().

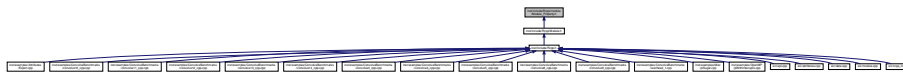
### 7.148.1.2 field\_readonly()

```
template<typename T >
self& field_readonly (
    const char * name_,
    T Class::* ptr,
    const char * docstring = 0 )
```

Definition at line 76 of file Module\_Field.h.

## 7.149 inst/include/Rcpp/module/Module\_Property.h File Reference

This graph shows which files directly or indirectly include this file:

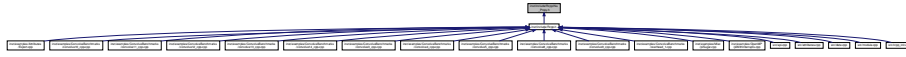


### Classes

- class CppProperty\_GetMethod< Class, PROP >
- class CppProperty\_GetConstMethod< Class, PROP >
- class CppProperty\_GetPointerMethod< Class, PROP >
- class CppProperty\_GetMethod\_SetMethod< Class, PROP >
- class CppProperty\_GetConstMethod\_SetMethod< Class, PROP >
- class CppProperty\_GetMethod\_SetPointer< Class, PROP >
- class CppProperty\_GetConstMethod\_SetPointer< Class, PROP >
- class CppProperty\_GetPointer\_SetMethod< Class, PROP >
- class CppProperty\_GetPointer\_SetPointer< Class, PROP >

## 7.150 inst/include/Rcpp/Na\_Proxy.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::Na\\_Proxy](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*

### Functions

- LogicalVector [Rcpp::shush\\_about\\_NA](#) ()

### Variables

- static Na\_Proxy [Rcpp::NA](#)

## 7.151 inst/include/Rcpp/Named.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::Argument](#)
- class [Rcpp::internal::NamedPlaceHolder](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Functions

- Argument [Rcpp::Named](#) (const std::string &name)
- `template<typename T >`  
`traits::named_object< T >` [Rcpp::Named](#) (const std::string &name, const T &o)

## Variables

- static internal::NamedPlaceholder [Rcpp::\\_](#)

## 7.152 inst/include/Rcpp/Nullable.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::Nullable< T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

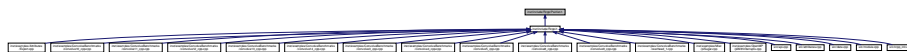
## 7.153 inst/include/Rcpp/Pairlist.h File Reference

```
#include <RcppCommon.h>
#include <Rcpp/DottedPair.h>
#include <Rcpp/r_cast.h>
#include <Rcpp/generated/Pairlist__ctors.h>
```

Include dependency graph for Pairlist.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Typedefs

- typedef Pairlist\_Impl< PreserveStorage > [Rcpp::Pairlist](#)

## Functions

- [Rcpp::RCPP\\_API\\_CLASS](#) (Pairlist\_Impl)
- [Rcpp::Pairlist\\_Impl](#) ()
- [Rcpp::Pairlist\\_Impl](#) (SEXP x)
- void [Rcpp::update](#) (SEXP)

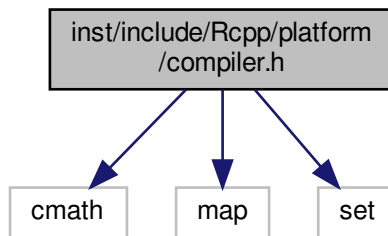
## Variables

- public [Rcpp::DottedPairProxyPolicy](#)< Pairlist\_Impl< StoragePolicy > >
- public public [Rcpp::DottedPairImpl](#)< Pairlist\_Impl< StoragePolicy > >

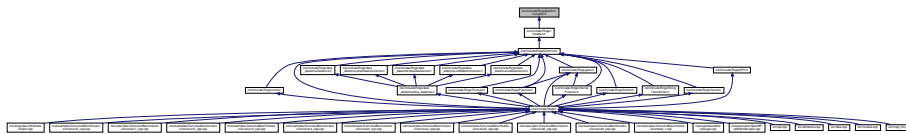
## 7.154 inst/include/Rcpp/platform/compiler.h File Reference

```
#include <cmath>
#include <map>
#include <set>
```

Include dependency graph for compiler.h:



This graph shows which files directly or indirectly include this file:



## Macros

- `#define RCPP_USING_MAP`
- `#define RCPP_UNORDERED_MAP std::map`
- `#define RCPP_USING_SET`
- `#define RCPP_UNORDERED_SET std::set`

### 7.154.1 Macro Definition Documentation

#### 7.154.1.1 RCPP\_UNORDERED\_MAP

```
#define RCPP_UNORDERED_MAP std::map
```

Definition at line 178 of file compiler.h.

#### 7.154.1.2 RCPP\_UNORDERED\_SET

```
#define RCPP_UNORDERED_SET std::set
```

Definition at line 187 of file compiler.h.

#### 7.154.1.3 RCPP\_USING\_MAP

```
#define RCPP_USING_MAP
```

Definition at line 177 of file compiler.h.

#### 7.154.1.4 RCPP\_USING\_SET

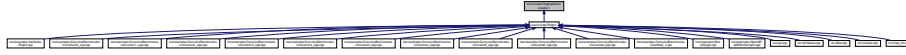
```
#define RCPP_USING_SET
```

Definition at line 186 of file compiler.h.



## 7.155 inst/include/Rcpp/platform/solaris.h File Reference

This graph shows which files directly or indirectly include this file:



## 7.156 inst/include/Rcpp/print.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

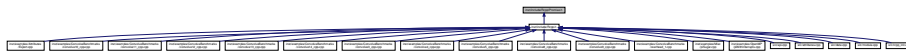
- [Rcpp](#)  
*Rcpp API.*

### Functions

- void [Rcpp::print](#) (SEXP s)
- void [Rcpp::warningcall](#) (SEXP call, const std::string &s)

## 7.157 inst/include/Rcpp/Promise.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Typedefs

- typedef Promise\_Impl< PreserveStorage > [Rcpp::Promise](#)

## Functions

- [Rcpp::RCPP\\_API\\_CLASS](#) (Promise\_Impl)

## 7.158 inst/include/Rcpp/protection/Armor.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::Armor< T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.159 inst/include/Rcpp/protection/Shelter.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::Shelter< T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.160 inst/include/Rcpp/protection/Shield.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::Shield< T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Functions

- SEXP [Rcpp::Rcpp\\_protect](#) (SEXP x)
- void [Rcpp::Rcpp\\_unprotect](#) (int i)

## 7.161 inst/include/Rcpp/proxy/AttributeProxy.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::AttributeProxyPolicy< CLASS >](#)
- class [Rcpp::AttributeProxyPolicy< CLASS >::AttributeProxy](#)
- class [Rcpp::AttributeProxyPolicy< CLASS >::const\\_AttributeProxy](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.162 inst/include/Rcpp/proxy/Binding.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

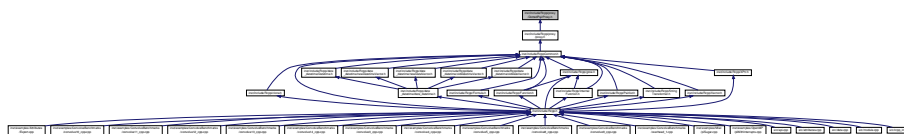
- class [Rcpp::BindingPolicy< EnvironmentClass >](#)
- class [Rcpp::BindingPolicy< EnvironmentClass >::Binding](#)
- class [Rcpp::BindingPolicy< EnvironmentClass >::const\\_Binding](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.163 inst/include/Rcpp/proxy/DottedPairProxy.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::DottedPairProxyPolicy< CLASS >](#)
- class [Rcpp::DottedPairProxyPolicy< CLASS >::DottedPairProxy](#)
- class [Rcpp::DottedPairProxyPolicy< CLASS >::const\\_DottedPairProxy](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.164 inst/include/Rcpp/proxy/FieldProxy.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::FieldProxyPolicy< CLASS >](#)
- class [Rcpp::FieldProxyPolicy< CLASS >::FieldProxy](#)
- class [Rcpp::FieldProxyPolicy< CLASS >::const\\_FieldProxy](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.165 inst/include/Rcpp/proxy/GenericProxy.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::GenericProxy< Proxy >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.166 inst/include/Rcpp/proxy/NamesProxy.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::NamesProxyPolicy< CLASS >](#)
- class [Rcpp::NamesProxyPolicy< CLASS >::NamesProxy](#)
- class [Rcpp::NamesProxyPolicy< CLASS >::const\\_NamesProxy](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.167 inst/include/Rcpp/proxy/ProtectedProxy.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::ProtectedProxyPolicy< XPtrClass >](#)
- class [Rcpp::ProtectedProxyPolicy< XPtrClass >::ProtectedProxy](#)
- class [Rcpp::ProtectedProxyPolicy< XPtrClass >::const\\_ProtectedProxy](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Functions

- [template<typename T >](#)  
[T Rcpp::as](#) (SEXP x)

## 7.168 inst/include/Rcpp/proxy/RObjectMethods.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::RObjectMethods< Class >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.169 inst/include/Rcpp/proxy/SlotProxy.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::SlotProxyPolicy< CLASS >](#)
- class [Rcpp::SlotProxyPolicy< CLASS >::SlotProxy](#)
- class [Rcpp::SlotProxyPolicy< CLASS >::const\\_SlotProxy](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.170 inst/include/Rcpp/proxy/TagProxy.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::TagProxyPolicy< XPtrClass >](#)
- class [Rcpp::TagProxyPolicy< XPtrClass >::TagProxy](#)
- class [Rcpp::TagProxyPolicy< XPtrClass >::const\\_TagProxy](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

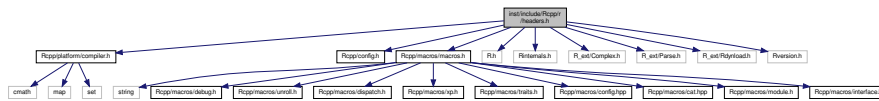
## 7.171 inst/include/Rcpp/r/headers.h File Reference

```
#include <Rcpp/platform/compiler.h>
#include <Rcpp/config.h>
#include <Rcpp/macros/macros.h>
#include <R.h>
#include <Rinternals.h>
#include <R_ext/Complex.h>
#include <R_ext/Parse.h>
#include <R_ext/Rdynload.h>
```



```
#include <Rversion.h>
```

Include dependency graph for headers.h:



This graph shows which files directly or indirectly include this file:



## Macros

- `#define MAXELTSIZE 8192`
- `#define R_NO_REMAP`
- `#define STRICT_R_HEADERS`
- `#define NORET`

### 7.171.1 Macro Definition Documentation

#### 7.171.1.1 MAXELTSIZE

```
#define MAXELTSIZE 8192
```

Definition at line 25 of file headers.h.

#### 7.171.1.2 NORET

```
#define NORET
```

Definition at line 78 of file headers.h.

### 7.171.1.3 R\_NO\_REMAP

```
#define R_NO_REMAP
```

Definition at line 26 of file headers.h.

### 7.171.1.4 STRICT\_R\_HEADERS

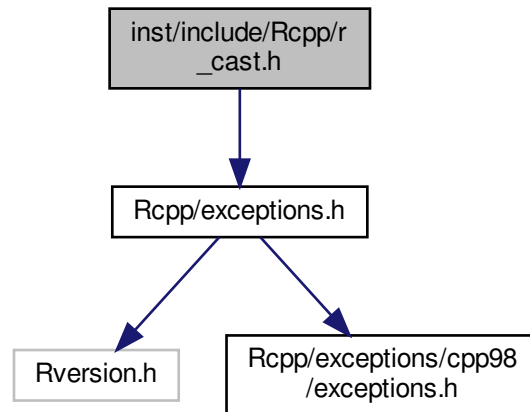
```
#define STRICT_R_HEADERS
```

Definition at line 31 of file headers.h.

## 7.172 inst/include/Rcpp/r\_cast.h File Reference

```
#include <Rcpp/exceptions.h>
```

Include dependency graph for r\_cast.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

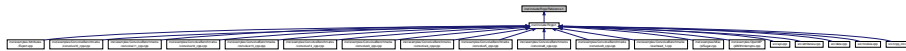
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Functions

- SEXP [Rcpp::internal::convert\\_using\\_rfunction](#) (SEXP x, const char \*const fun)
- `template<int TARGET>`  
SEXP [Rcpp::internal::r\\_true\\_cast](#) (SEXP x)
- `template<int RTYPE>`  
SEXP [Rcpp::internal::basic\\_cast](#) (SEXP x)
- `template<>` SEXP [Rcpp::internal::r\\_true\\_cast< INTSXP >](#) (SEXP x)
- `template<>` SEXP [Rcpp::internal::r\\_true\\_cast< REALSXP >](#) (SEXP x)
- `template<>` SEXP [Rcpp::internal::r\\_true\\_cast< RAWSXP >](#) (SEXP x)
- `template<>` SEXP [Rcpp::internal::r\\_true\\_cast< CPLXSXP >](#) (SEXP x)
- `template<>` SEXP [Rcpp::internal::r\\_true\\_cast< LGLSXP >](#) (SEXP x)
- `template<>` SEXP [Rcpp::internal::r\\_true\\_cast< STRSXP >](#) (SEXP x)
- `template<>` SEXP [Rcpp::internal::r\\_true\\_cast< VECSXP >](#) (SEXP x)
- `template<>` SEXP [Rcpp::internal::r\\_true\\_cast< EXPRSXP >](#) (SEXP x)
- `template<>` SEXP [Rcpp::internal::r\\_true\\_cast< LISTSXP >](#) (SEXP x)
- `template<>` SEXP [Rcpp::internal::r\\_true\\_cast< LANGSXP >](#) (SEXP x)
- `template<int TARGET>`  
SEXP [Rcpp::r\\_cast](#) (SEXP x)

## 7.173 inst/include/Rcpp/Reference.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Typedefs

- typedef `Reference_Impl< PreserveStorage >` [Rcpp::Reference](#)

## Functions

- [Rcpp::RCPP\\_API\\_CLASS](#) (Reference\_Impl)

## Variables

- public [Rcpp::FieldProxyPolicy](#)< [Reference\\_Impl](#)< [StoragePolicy](#) > >

## 7.174 inst/include/Rcpp/Rmath.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [R](#)

## Functions

- double [R::norm\\_rand](#) (void)
- double [R::unif\\_rand](#) (void)
- double [R::exp\\_rand](#) (void)
- double [R::dnorm](#) (double x, double mu, double sigma, int lg)
- double [R::pnorm](#) (double x, double mu, double sigma, int lt, int lg)
- double [R::qnorm](#) (double p, double mu, double sigma, int lt, int lg)
- double [R::rnorm](#) (double mu, double sigma)
- void [R::pnorm\\_both](#) (double x, double \*cum, double \*ccum, int lt, int lg)
- double [R::dunif](#) (double x, double a, double b, int lg)
- double [R::punif](#) (double x, double a, double b, int lt, int lg)
- double [R::qunif](#) (double p, double a, double b, int lt, int lg)
- double [R::runif](#) (double a, double b)
- double [R::dgamma](#) (double x, double shp, double scl, int lg)
- double [R::pgamma](#) (double x, double alp, double scl, int lt, int lg)
- double [R::qgamma](#) (double p, double alp, double scl, int lt, int lg)
- double [R::rgamma](#) (double a, double scl)
- double [R::log1pmx](#) (double x)
- double [R::log1pexp](#) (double x)
- double [R::lgamma1p](#) (double a)
- double [R::logspace\\_add](#) (double lx, double ly)
- double [R::logspace\\_sub](#) (double lx, double ly)
- double [R::dbeta](#) (double x, double a, double b, int lg)

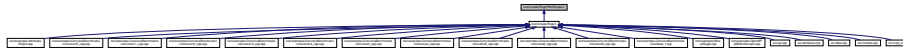
- double [R::pbeta](#) (double x, double p, double q, int lt, int lg)
- double [R::qbeta](#) (double a, double p, double q, int lt, int lg)
- double [R::rbeta](#) (double a, double b)
- double [R::dlnorm](#) (double x, double ml, double sl, int lg)
- double [R::plnorm](#) (double x, double ml, double sl, int lt, int lg)
- double [R::qlnorm](#) (double p, double ml, double sl, int lt, int lg)
- double [R::rlnorm](#) (double ml, double sl)
- double [R::dchisq](#) (double x, double df, int lg)
- double [R::pchisq](#) (double x, double df, int lt, int lg)
- double [R::qchisq](#) (double p, double df, int lt, int lg)
- double [R::rchisq](#) (double df)
- double [R::dnchisq](#) (double x, double df, double ncp, int lg)
- double [R::pnchisq](#) (double x, double df, double ncp, int lt, int lg)
- double [R::qnchisq](#) (double p, double df, double ncp, int lt, int lg)
- double [R::rnchisq](#) (double df, double lb)
- double [R::df](#) (double x, double df1, double df2, int lg)
- double [R::pf](#) (double x, double df1, double df2, int lt, int lg)
- double [R::qf](#) (double p, double df1, double df2, int lt, int lg)
- double [R::rf](#) (double df1, double df2)
- double [R::dt](#) (double x, double n, int lg)
- double [R::pt](#) (double x, double n, int lt, int lg)
- double [R::qt](#) (double p, double n, int lt, int lg)
- double [R::rt](#) (double n)
- double [R::dbinom](#) (double x, double n, double p, int lg)
- double [R::pbinom](#) (double x, double n, double p, int lt, int lg)
- double [R::qbinom](#) (double p, double n, double m, int lt, int lg)
- double [R::rbinom](#) (double n, double p)
- void [R::rmultinom](#) (int n, double \*prob, int k, int \*rn)
- double [R::dcauchy](#) (double x, double lc, double sl, int lg)
- double [R::pcauchy](#) (double x, double lc, double sl, int lt, int lg)
- double [R::qcauchy](#) (double p, double lc, double sl, int lt, int lg)
- double [R::rcauchy](#) (double lc, double sl)
- double [R::dexp](#) (double x, double sl, int lg)
- double [R::pexp](#) (double x, double sl, int lt, int lg)
- double [R::qexp](#) (double p, double sl, int lt, int lg)
- double [R::rexp](#) (double sl)
- double [R::dgeom](#) (double x, double p, int lg)
- double [R::pgeom](#) (double x, double p, int lt, int lg)
- double [R::qgeom](#) (double p, double pb, int lt, int lg)
- double [R::rgeom](#) (double p)
- double [R::dhyper](#) (double x, double r, double b, double n, int lg)
- double [R::phyper](#) (double x, double r, double b, double n, int lt, int lg)
- double [R::qhyper](#) (double p, double r, double b, double n, int lt, int lg)
- double [R::rhyper](#) (double r, double b, double n)
- double [R::dnbinom](#) (double x, double sz, double pb, int lg)
- double [R::pnbinom](#) (double x, double sz, double pb, int lt, int lg)
- double [R::qnbinom](#) (double p, double sz, double pb, int lt, int lg)
- double [R::rnbinom](#) (double sz, double pb)
- double [R::dnbinom\\_mu](#) (double x, double sz, double mu, int lg)
- double [R::pnbinom\\_mu](#) (double x, double sz, double mu, int lt, int lg)
- double [R::qnbinom\\_mu](#) (double x, double sz, double mu, int lt, int lg)

- double [R::dpois](#) (double x, double lb, int lg)
- double [R::ppois](#) (double x, double lb, int lt, int lg)
- double [R::qpois](#) (double p, double lb, int lt, int lg)
- double [R::rpois](#) (double mu)
- double [R::dweibull](#) (double x, double sh, double sl, int lg)
- double [R::pweibull](#) (double x, double sh, double sl, int lt, int lg)
- double [R::qweibull](#) (double p, double sh, double sl, int lt, int lg)
- double [R::rweibull](#) (double sh, double sl)
- double [R::dlogis](#) (double x, double lc, double sl, int lg)
- double [R::plogis](#) (double x, double lc, double sl, int lt, int lg)
- double [R::qlogis](#) (double p, double lc, double sl, int lt, int lg)
- double [R::rlogis](#) (double lc, double sl)
- double [R::dnbeta](#) (double x, double a, double b, double ncp, int lg)
- double [R::pnbeta](#) (double x, double a, double b, double ncp, int lt, int lg)
- double [R::qnbeta](#) (double p, double a, double b, double ncp, int lt, int lg)
- double [R::rnbeta](#) (double a, double b, double np)
- double [R::dnf](#) (double x, double df1, double df2, double ncp, int lg)
- double [R::pnf](#) (double x, double df1, double df2, double ncp, int lt, int lg)
- double [R::qnf](#) (double p, double df1, double df2, double ncp, int lt, int lg)
- double [R::dnt](#) (double x, double df, double ncp, int lg)
- double [R::pnt](#) (double x, double df, double ncp, int lt, int lg)
- double [R::qnt](#) (double p, double df, double ncp, int lt, int lg)
- double [R::ptukey](#) (double q, double rr, double cc, double df, int lt, int lg)
- double [R::qtukey](#) (double p, double rr, double cc, double df, int lt, int lg)
- double [R::dwilcox](#) (double x, double m, double n, int lg)
- double [R::pwilcox](#) (double q, double m, double n, int lt, int lg)
- double [R::qwilcox](#) (double x, double m, double n, int lt, int lg)
- double [R::rwilcox](#) (double m, double n)
- double [R::dsignrank](#) (double x, double n, int lg)
- double [R::psignrank](#) (double x, double n, int lt, int lg)
- double [R::qsignrank](#) (double x, double n, int lt, int lg)
- double [R::rsignrank](#) (double n)
- double [R::gammafn](#) (double x)
- double [R::lgammafn](#) (double x)
- double [R::lgammafn\\_sign](#) (double x, int \*sgn)
- void [R::dpsifn](#) (double x, int n, int kode, int m, double \*ans, int \*nz, int \*ierr)
- double [R::psigamma](#) (double x, double deriv)
- double [R::digamma](#) (double x)
- double [R::trigamma](#) (double x)
- double [R::tetragamma](#) (double x)
- double [R::pentagamma](#) (double x)
- double [R::beta](#) (double a, double b)
- double [R::lbeta](#) (double a, double b)
- double [R::choose](#) (double n, double k)
- double [R::lchoose](#) (double n, double k)
- double [R::bessel\\_i](#) (double x, double al, double ex)
- double [R::bessel\\_j](#) (double x, double al)
- double [R::bessel\\_k](#) (double x, double al, double ex)
- double [R::bessel\\_y](#) (double x, double al)
- double [R::bessel\\_i\\_ex](#) (double x, double al, double ex, double \*bi)
- double [R::bessel\\_j\\_ex](#) (double x, double al, double \*bj)

- double [R::bessel\\_k\\_ex](#) (double x, double a1, double ex, double \*bk)
- double [R::bessel\\_y\\_ex](#) (double x, double a1, double \*by)
- double [R::hypot](#) (double a, double b)
- double [R::expm1](#) (double x)
- double [R::log1p](#) (double x)
- int [R::imax2](#) (int x, int y)
- int [R::imin2](#) (int x, int y)
- double [R::fmax2](#) (double x, double y)
- double [R::fmin2](#) (double x, double y)
- double [R::sign](#) (double x)
- double [R::fprec](#) (double x, double dg)
- double [R::fround](#) (double x, double dg)
- double [R::fsign](#) (double x, double y)
- double [R::ftrunc](#) (double x)

## 7.175 inst/include/Rcpp/RNGScope.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

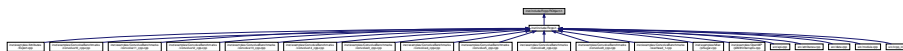
- class [Rcpp::RNGScope](#)
- class [Rcpp::SuspendRNGSynchronizationScope](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.176 inst/include/Rcpp/RObject.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Typedefs

- typedef RObject\_Impl< PreserveStorage > [Rcpp::RObject](#)

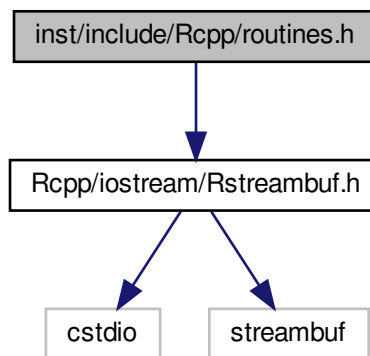
## Functions

- [Rcpp::RCPP\\_API\\_CLASS](#) (RObject\_Impl)

## 7.177 inst/include/Rcpp/routines.h File Reference

```
#include <Rcpp/iostream/Rstreambuf.h>
```

Include dependency graph for routines.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*



## Macros

- #define [GET\\_CALLABLE](#)(\_\_FUN\_\_) (Fun) R\_GetCCallable( "Rcpp", \_\_FUN\_\_ )

## Functions

- attribute\_hidden const char \* [Rcpp::type2name](#) (SEXP x)
- attribute\_hidden unsigned long [Rcpp::internal::enterRNGScope](#) ()
- attribute\_hidden unsigned long [Rcpp::internal::exitRNGScope](#) ()
- attribute\_hidden unsigned long [Rcpp::internal::beginSuspendRNGSynchronization](#) ()
- attribute\_hidden unsigned long [Rcpp::internal::endSuspendRNGSynchronization](#) ()
- attribute\_hidden char \* [Rcpp::internal::get\\_string\\_buffer](#) ()
- attribute\_hidden SEXP [Rcpp::internal::get\\_Rcpp\\_namespace](#) ()
- attribute\_hidden double [Rcpp::mktime00](#) (struct tm &tm)
- attribute\_hidden struct tm \* [Rcpp::gmtime\\_](#) (const time\_t \*const x)
- attribute\_hidden void [Rcpp::Rcpp\\_precious\\_init](#) ()
- attribute\_hidden void [Rcpp::Rcpp\\_precious\\_teardown](#) ()
- attribute\_hidden SEXP [Rcpp::Rcpp\\_precious\\_preserve](#) (SEXP object)
- attribute\_hidden void [Rcpp::Rcpp\\_precious\\_remove](#) (SEXP token)
- attribute\_hidden Rostream< true > & [Rcpp::Rcpp\\_cout\\_get](#) ()
- attribute\_hidden Rostream< false > & [Rcpp::Rcpp\\_cerr\\_get](#) ()
- attribute\_hidden SEXP [rcpp\\_get\\_stack\\_trace](#) ()
- attribute\_hidden SEXP [rcpp\\_set\\_stack\\_trace](#) (SEXP e)
- attribute\_hidden std::string [demangle](#) (const std::string &name)
- attribute\_hidden const char \* [short\\_file\\_name](#) (const char \*file)
- attribute\_hidden SEXP [stack\\_trace](#) (const char \*file="", int line=-1)
- attribute\_hidden SEXP [get\\_string\\_elt](#) (SEXP s, R\_xlen\_t i)
- attribute\_hidden const char \* [char\\_get\\_string\\_elt](#) (SEXP s, R\_xlen\_t i)
- attribute\_hidden void [set\\_string\\_elt](#) (SEXP s, R\_xlen\_t i, SEXP v)
- attribute\_hidden void [char\\_set\\_string\\_elt](#) (SEXP s, R\_xlen\_t i, const char \*v)
- attribute\_hidden SEXP \* [get\\_string\\_ptr](#) (SEXP s)
- attribute\_hidden SEXP [get\\_vector\\_elt](#) (SEXP v, R\_xlen\_t i)
- attribute\_hidden void [set\\_vector\\_elt](#) (SEXP v, R\_xlen\_t i, SEXP x)
- attribute\_hidden SEXP \* [get\\_vector\\_ptr](#) (SEXP v)
- attribute\_hidden const char \* [char\\_nocheck](#) (SEXP x)
- attribute\_hidden void \* [dataptr](#) (SEXP x)
- attribute\_hidden [Rcpp::Module](#) \* [getCurrentScope](#) ()
- attribute\_hidden void [setCurrentScope](#) ([Rcpp::Module](#) \*mod)
- attribute\_hidden int \* [get\\_cache](#) (int n)
- attribute\_hidden SEXP [reset\\_current\\_error](#) ()
- attribute\_hidden int [error\\_occured](#) ()
- attribute\_hidden SEXP [rcpp\\_get\\_current\\_error](#) ()

### 7.177.1 Macro Definition Documentation

### 7.177.1.1 GET\_CALLABLE

```
#define GET_CALLABLE(  
    __FUN__ ) (Fun) R_GetCCallable( "Rcpp", __FUN__ )
```

Definition at line 82 of file routines.h.

## 7.177.2 Function Documentation

### 7.177.2.1 char\_get\_string\_elt()

```
attribute_hidden const char* char_get_string_elt (  
    SEXP s,  
    R_xlen_t i ) [inline]
```

Definition at line 216 of file routines.h.

### 7.177.2.2 char\_nocheck()

```
attribute_hidden const char* char_nocheck (  
    SEXP x ) [inline]
```

Definition at line 258 of file routines.h.

### 7.177.2.3 char\_set\_string\_elt()

```
attribute_hidden void char_set_string_elt (  
    SEXP s,  
    R_xlen_t i,  
    const char * v ) [inline]
```

Definition at line 228 of file routines.h.

#### 7.177.2.4 `dataptr()`

```
attribute_hidden void* dataptr (  
    SEXP x ) [inline]
```

Definition at line 264 of file routines.h.

#### 7.177.2.5 `demangle()`

```
attribute_hidden std::string demangle (  
    const std::string & name ) [inline]
```

Definition at line 192 of file routines.h.

#### 7.177.2.6 `error_occured()`

```
attribute_hidden int error_occured ( ) [inline]
```

Definition at line 294 of file routines.h.

References `GET_CALLABLE`.

Referenced by `registerFunctions()`.

#### 7.177.2.7 `get_cache()`

```
attribute_hidden int* get_cache (  
    int n ) [inline]
```

Definition at line 282 of file routines.h.

References `GET_CALLABLE`.

Referenced by `Rcpp::sugar::IndexHash< RTYPE >::IndexHash()`, and `registerFunctions()`.

### 7.177.2.8 `get_string_elt()`

```
attribute_hidden SEXP get_string_elt (  
    SEXP s,  
    R_xlen_t i ) [inline]
```

Definition at line 210 of file routines.h.

### 7.177.2.9 `get_string_ptr()`

```
attribute_hidden SEXP* get_string_ptr (  
    SEXP s ) [inline]
```

Definition at line 234 of file routines.h.

### 7.177.2.10 `get_vector_elt()`

```
attribute_hidden SEXP get_vector_elt (  
    SEXP v,  
    R_xlen_t i ) [inline]
```

Definition at line 240 of file routines.h.

### 7.177.2.11 `get_vector_ptr()`

```
attribute_hidden SEXP* get_vector_ptr (  
    SEXP v ) [inline]
```

Definition at line 252 of file routines.h.

### 7.177.2.12 `getCurrentScope()`

```
attribute_hidden Rcpp::Module* getCurrentScope ( ) [inline]
```

Definition at line 270 of file routines.h.

References `GET_CALLABLE`.

Referenced by `class_< Class >::derives()`, `class_< Class >::get_instance()`, `registerFunctions()`, and `Rcpp::enum_< Enum, Parent >::~~enum_()`.

### 7.177.2.13 `rcpp_get_current_error()`

```
attribute_hidden SEXP rcpp_get_current_error ( ) [inline]
```

Definition at line 300 of file routines.h.

References `GET_CALLABLE`.

Referenced by `registerFunctions()`.

### 7.177.2.14 `rcpp_get_stack_trace()`

```
attribute_hidden SEXP rcpp_get_stack_trace ( ) [inline]
```

Definition at line 180 of file routines.h.

References `GET_CALLABLE`.

Referenced by `exception_to_condition_template()`, and `registerFunctions()`.

### 7.177.2.15 `rcpp_set_stack_trace()`

```
attribute_hidden SEXP rcpp_set_stack_trace (  
    SEXP e ) [inline]
```

Definition at line 186 of file routines.h.

References `GET_CALLABLE`.

Referenced by `Rcpp::exception::copy_stack_trace_to_r()`, `exception_to_condition_template()`, and `registerFunctions()`.

### 7.177.2.16 `reset_current_error()`

```
attribute_hidden SEXP reset_current_error ( ) [inline]
```

Definition at line 288 of file routines.h.

References `GET_CALLABLE`.

Referenced by `registerFunctions()`.

### 7.177.2.17 `set_string_elt()`

```
attribute_hidden void set_string_elt (
    SEXP s,
    R_xlen_t i,
    SEXP v ) [inline]
```

Definition at line 222 of file routines.h.

### 7.177.2.18 `set_vector_elt()`

```
attribute_hidden void set_vector_elt (
    SEXP v,
    R_xlen_t i,
    SEXP x ) [inline]
```

Definition at line 246 of file routines.h.

### 7.177.2.19 `setCurrentScope()`

```
attribute_hidden void setCurrentScope (
    Rcpp::Module * mod ) [inline]
```

Definition at line 276 of file routines.h.

References `GET_CALLABLE`.

Referenced by `R_init_Rcpp()`, and `registerFunctions()`.

### 7.177.2.20 `short_file_name()`

```
attribute_hidden const char* short_file_name (
    const char * file ) [inline]
```

Definition at line 198 of file routines.h.

References `GET_CALLABLE`.

7.177.2.21 `stack_trace()`

```
attribute_hidden SEXP stack_trace (
    const char * file = "",
    int line = -1 ) [inline]
```

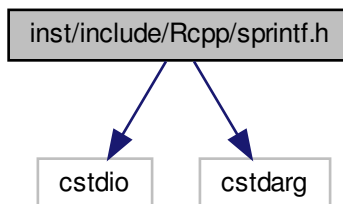
Definition at line 204 of file routines.h.

References `GET_CALLABLE`.

Referenced by `registerFunctions()`.

## 7.178 inst/include/Rcpp/sprintf.h File Reference

```
#include <cstdio>
#include <cstdarg>
Include dependency graph for sprintf.h:
```



This graph shows which files directly or indirectly include this file:



## Namespaces

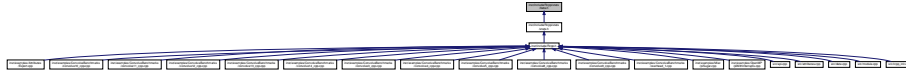
- [Rcpp](#)  
*Rcpp API.*

## Functions

- `template<int MAX_SIZE>`  
`std::string Rcpp::sprintf (const char *format,...)`

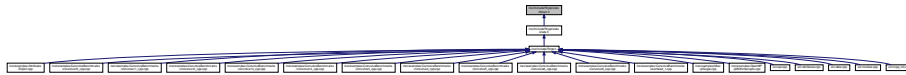
### 7.179 inst/include/Rcpp/stats/beta.h File Reference

This graph shows which files directly or indirectly include this file:



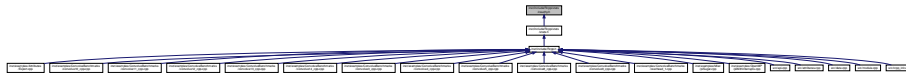
### 7.180 inst/include/Rcpp/stats/binom.h File Reference

This graph shows which files directly or indirectly include this file:



### 7.181 inst/include/Rcpp/stats/cauchy.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

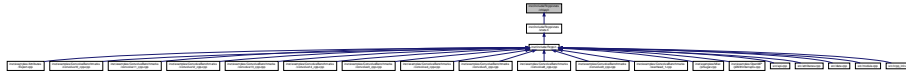
## Functions

- double [Rcpp::stats::dcauchy\\_0](#) (double x, int [give\\_log](#))
- double [Rcpp::stats::dcauchy\\_1](#) (double x, double location, int [give\\_log](#))
- double [Rcpp::stats::pcauchy\\_0](#) (double x, int lower\_tail, int log\_p)
- double [Rcpp::stats::pcauchy\\_1](#) (double x, double location, int lower\_tail, int log\_p)
- double [Rcpp::stats::qcauchy\\_0](#) (double p, int lower\_tail, int log\_p)
- double [Rcpp::stats::qcauchy\\_1](#) (double p, double location, int lower\_tail, int log\_p)



## 7.182 inst/include/Rcpp/stats/chisq.h File Reference

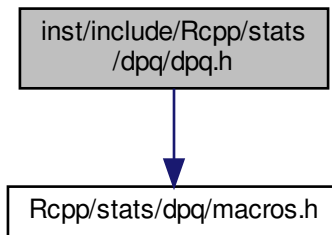
This graph shows which files directly or indirectly include this file:



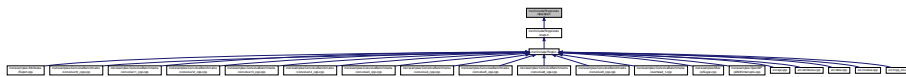
## 7.183 inst/include/Rcpp/stats/dpq/dpq.h File Reference

```
#include <Rcpp/stats/dpq/macros.h>
```

Include dependency graph for dpq.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::stats::D0](#)< RTYPE, NA, T >
- class [Rcpp::stats::D1](#)< RTYPE, NA, T >
- class [Rcpp::stats::D2](#)< RTYPE, NA, T >
- class [Rcpp::stats::D3](#)< RTYPE, NA, T >
- class [Rcpp::stats::P0](#)< RTYPE, NA, T >
- class [Rcpp::stats::P1](#)< RTYPE, NA, T >
- class [Rcpp::stats::P2](#)< RTYPE, NA, T >
- class [Rcpp::stats::P3](#)< RTYPE, NA, T >
- class [Rcpp::stats::Q0](#)< RTYPE, NA, T >
- class [Rcpp::stats::Q1](#)< RTYPE, NA, T >
- class [Rcpp::stats::Q2](#)< RTYPE, NA, T >
- class [Rcpp::stats::Q3](#)< RTYPE, NA, T >



## 7.183.1.2 RCPP\_DPQ\_1

```
#define RCPP_DPQ_1(
    __NAME__,
    __D__,
    __P__,
    __Q__ )
```

**Value:**

```
namespace Rcpp {
template <int RTYPE, bool NA, typename T>
inline stats::D1<RTYPE,NA,T> d##__NAME__(
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, bool log = false
) {
    return stats::D1<RTYPE,NA,T>( __D__, x, p0, log );
}
template <int RTYPE, bool NA, typename T>
inline stats::P1<RTYPE,NA,T> p##__NAME__(
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, bool lower = true, bool log = false \
) {
    return stats::P1<RTYPE,NA,T>( __P__, x, p0, lower, log );
}
template <int RTYPE, bool NA, typename T>
inline stats::Q1<RTYPE,NA,T> q##__NAME__(
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, bool lower = true, bool log = false \
) {
    return stats::Q1<RTYPE,NA,T>( __Q__, x, p0, lower, log );
} }
} }
```

Definition at line 340 of file dpq.h.

## 7.183.1.3 RCPP\_DPQ\_2

```
#define RCPP_DPQ_2(
    __NAME__,
    __D__,
    __P__,
    __Q__ )
```

**Value:**

```
namespace Rcpp {
template <int RTYPE, bool NA, typename T>
inline stats::D2<RTYPE,NA,T> d##__NAME__(
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, double p1, bool log = false
) {
    return stats::D2<RTYPE,NA,T>( __D__, x, p0, p1, log );
}
template <int RTYPE, bool NA, typename T>
inline stats::P2<RTYPE,NA,T> p##__NAME__(
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, double p1, bool lower = true, bool log = false \
) {
    return stats::P2<RTYPE,NA,T>( __P__, x, p0, p1, lower, log );
}
template <int RTYPE, bool NA, typename T>
inline stats::Q2<RTYPE,NA,T> q##__NAME__(
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, double p1, bool lower = true, bool log = false \
) {
    return stats::Q2<RTYPE,NA,T>( __Q__, x, p0, p1, lower, log );
} }
} }
```

Definition at line 363 of file dpq.h.

### 7.183.1.4 RCPP\_DPQ\_3

```
#define RCPP_DPQ_3(
    __NAME__,
    __D__,
    __P__,
    __Q__ )
```

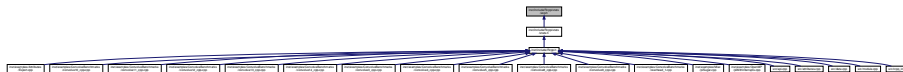
#### Value:

```
namespace Rcpp {
template <int RTYPE, bool NA, typename T>
inline stats::D3<RTYPE,NA,T> d##__NAME__(
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, double p1, double p2, bool log = false
) {
    return stats::D3<RTYPE,NA,T>( __D__, x, p0, p1, p2, log );
}
template <int RTYPE, bool NA, typename T>
inline stats::P3<RTYPE,NA,T> p##__NAME__(
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, double p1, double p2, bool lower = true, bool log = false
) {
    return stats::P3<RTYPE,NA,T>( __P__, x, p0, p1, p2, lower, log );
}
template <int RTYPE, bool NA, typename T>
inline stats::Q3<RTYPE,NA,T> q##__NAME__(
    const Rcpp::VectorBase<RTYPE,NA,T>& x, double p0, double p1, double p2, bool lower = true, bool log = false
) {
    return stats::Q3<RTYPE,NA,T>( __Q__, x, p0, p1, p2, lower, log );
} }
```

Definition at line 386 of file dpq.h.

## 7.184 inst/include/Rcpp/stats/exp.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

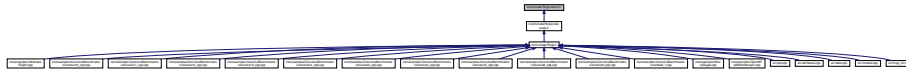
- [Rcpp](#)
  - Rcpp API.*
- [Rcpp::stats](#)

## Functions

- double [Rcpp::stats::d\\_exp\\_0](#) (double x, int [give\\_log](#))
- double [Rcpp::stats::q\\_exp\\_0](#) (double p, int lower\_tail, int log\_p)
- double [Rcpp::stats::p\\_exp\\_0](#) (double x, int lower\_tail, int log\_p)
- template<bool NA, typename T >  
stats::D1< REALSXP, NA, T > [Rcpp::dexp](#) (const [Rcpp::VectorBase](#)< REALSXP, NA, T > &x, double shape, bool log=false)
- template<bool NA, typename T >  
stats::P1< REALSXP, NA, T > [Rcpp::pexp](#) (const [Rcpp::VectorBase](#)< REALSXP, NA, T > &x, double shape, bool lower=true, bool log=false)
- template<bool NA, typename T >  
stats::Q1< REALSXP, NA, T > [Rcpp::qexp](#) (const [Rcpp::VectorBase](#)< REALSXP, NA, T > &x, double shape, bool lower=true, bool log=false)

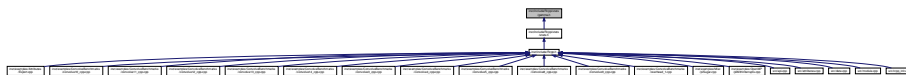
## 7.185 inst/include/Rcpp/stats/f.h File Reference

This graph shows which files directly or indirectly include this file:



## 7.186 inst/include/Rcpp/stats/gamma.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

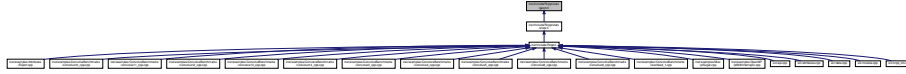
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## Functions

- double [Rcpp::stats::dgamma\\_1](#) (double x, double shape, int log\_p)
- double [Rcpp::stats::pgamma\\_1](#) (double x, double alpha, int lower\_tail, int log\_p)
- double [Rcpp::stats::qgamma\\_1](#) (double p, double alpha, int lower\_tail, int log\_p)

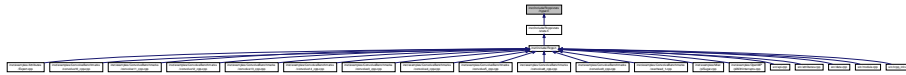
## 7.187 inst/include/Rcpp/stats/geom.h File Reference

This graph shows which files directly or indirectly include this file:



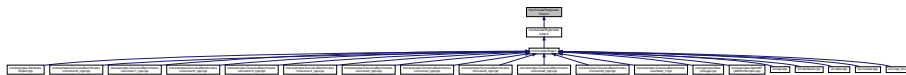
## 7.188 inst/include/Rcpp/stats/hyper.h File Reference

This graph shows which files directly or indirectly include this file:



## 7.189 inst/include/Rcpp/stats/lnorm.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

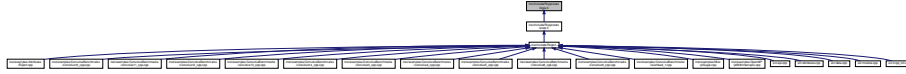
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## Functions

- double [Rcpp::stats::dlnorm\\_0](#) (double x, int log\_p)
- double [Rcpp::stats::dlnorm\\_1](#) (double x, double meanlog, int log\_p)
- double [Rcpp::stats::plnorm\\_0](#) (double x, int lower\_tail, int log\_p)
- double [Rcpp::stats::plnorm\\_1](#) (double x, double meanlog, int lower\_tail, int log\_p)
- double [Rcpp::stats::qlnorm\\_0](#) (double p, int lower\_tail, int log\_p)
- double [Rcpp::stats::qlnorm\\_1](#) (double p, double meanlog, int lower\_tail, int log\_p)

## 7.190 inst/include/Rcpp/stats/logis.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

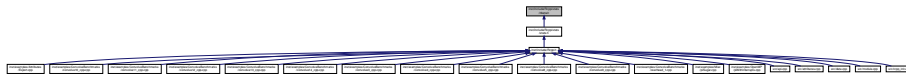
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

### Functions

- double [Rcpp::stats::dlogis\\_0](#) (double x, int [give\\_log](#))
- double [Rcpp::stats::dlogis\\_1](#) (double x, double location, int [give\\_log](#))
- double [Rcpp::stats::plogis\\_0](#) (double x, int lower\_tail, int log\_p)
- double [Rcpp::stats::plogis\\_1](#) (double x, double location, int lower\_tail, int log\_p)
- double [Rcpp::stats::qlogis\\_0](#) (double p, int lower\_tail, int log\_p)
- double [Rcpp::stats::qlogis\\_1](#) (double p, double location, int lower\_tail, int log\_p)

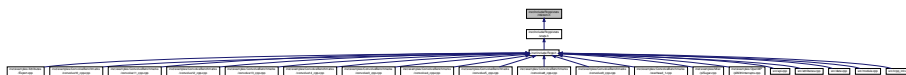
## 7.191 inst/include/Rcpp/stats/nbeta.h File Reference

This graph shows which files directly or indirectly include this file:



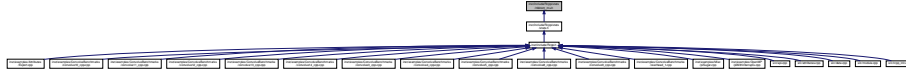
## 7.192 inst/include/Rcpp/stats/nbinom.h File Reference

This graph shows which files directly or indirectly include this file:



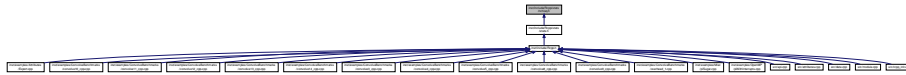
### 7.193 `inst/include/Rcpp/stats/nbinom_mu.h` File Reference

This graph shows which files directly or indirectly include this file:



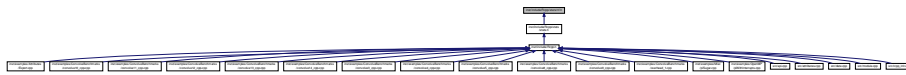
### 7.194 `inst/include/Rcpp/stats/nchisq.h` File Reference

This graph shows which files directly or indirectly include this file:



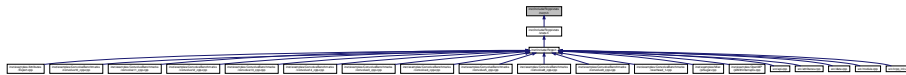
### 7.195 `inst/include/Rcpp/stats/nf.h` File Reference

This graph shows which files directly or indirectly include this file:



### 7.196 `inst/include/Rcpp/stats/norm.h` File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

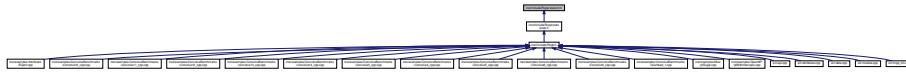


## Functions

- double [Rcpp::stats::dnorm\\_1](#) (double x, double mu, int [give\\_log](#))
- double [Rcpp::stats::dnorm\\_0](#) (double x, int [give\\_log](#))
- double [Rcpp::stats::pnorm\\_1](#) (double x, double mu, int lower\_tail, int log\_p)
- double [Rcpp::stats::pnorm\\_0](#) (double x, int lower\_tail, int log\_p)
- double [Rcpp::stats::qnorm\\_1](#) (double p, double mu, int lower\_tail, int log\_p)
- double [Rcpp::stats::qnorm\\_0](#) (double p, int lower\_tail, int log\_p)

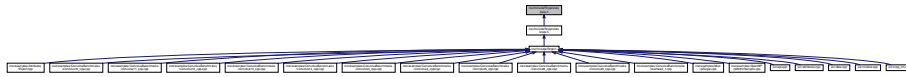
## 7.197 inst/include/Rcpp/stats/nt.h File Reference

This graph shows which files directly or indirectly include this file:



## 7.198 inst/include/Rcpp/stats/pois.h File Reference

This graph shows which files directly or indirectly include this file:



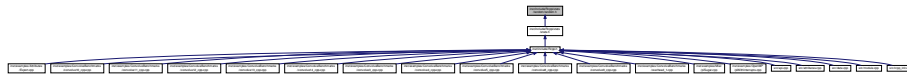
## 7.199 inst/include/Rcpp/stats/random/random.h File Reference

```
#include <Rcpp/stats/random/rnorm.h>
#include <Rcpp/stats/random/runif.h>
#include <Rcpp/stats/random/rgamma.h>
#include <Rcpp/stats/random/rbeta.h>
#include <Rcpp/stats/random/rlnorm.h>
#include <Rcpp/stats/random/rchisq.h>
#include <Rcpp/stats/random/rnchisq.h>
#include <Rcpp/stats/random/rf.h>
#include <Rcpp/stats/random/rt.h>
#include <Rcpp/stats/random/rbinom.h>
#include <Rcpp/stats/random/rcauchy.h>
#include <Rcpp/stats/random/rexp.h>
#include <Rcpp/stats/random/rgeom.h>
#include <Rcpp/stats/random/rnbinom.h>
#include <Rcpp/stats/random/rnbinom_mu.h>
#include <Rcpp/stats/random/rpois.h>
#include <Rcpp/stats/random/rweibull.h>
```

```
#include <Rcpp/stats/random/rlogis.h>
#include <Rcpp/stats/random/rwilcox.h>
#include <Rcpp/stats/random/rsignrank.h>
#include <Rcpp/stats/random/rhyper.h>
Include dependency graph for random.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::Generator< T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

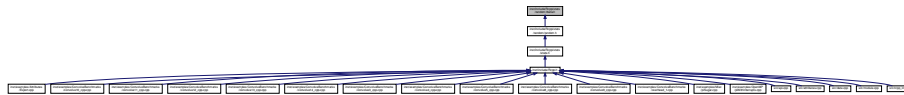
## Functions

- NumericVector [Rcpp::rnorm](#) (int n, double mean, double sd)
- NumericVector [Rcpp::rnorm](#) (int n, double mean)
- NumericVector [Rcpp::rnorm](#) (int n)
- NumericVector [Rcpp::rbeta](#) (int n, double a, double b)
- NumericVector [Rcpp::rbinom](#) (int n, double nin, double pp)
- NumericVector [Rcpp::rcauchy](#) (int n, double location, double scale)
- NumericVector [Rcpp::rcauchy](#) (int n, double location)
- NumericVector [Rcpp::rcauchy](#) (int n)
- NumericVector [Rcpp::rchisq](#) (int n, double df)
- NumericVector [Rcpp::rexp](#) (int n, double rate)
- NumericVector [Rcpp::rexp](#) (int n)
- NumericVector [Rcpp::rf](#) (int n, double n1, double n2)
- NumericVector [Rcpp::rgamma](#) (int n, double a, double scale)
- NumericVector [Rcpp::rgamma](#) (int n, double a)
- NumericVector [Rcpp::rgeom](#) (int n, double p)
- NumericVector [Rcpp::rhyper](#) (int n, double nn1, double nn2, double kk)
- NumericVector [Rcpp::rlnorm](#) (int n, double meanlog, double sdlog)
- NumericVector [Rcpp::rlnorm](#) (int n, double meanlog)
- NumericVector [Rcpp::rlnorm](#) (int n)
- NumericVector [Rcpp::rlogis](#) (int n, double location, double scale)

- NumericVector [Rcpp::rlogis](#) (int n, double location)
- NumericVector [Rcpp::rlogis](#) (int n)
- NumericVector [Rcpp::rnbinom](#) (int n, double siz, double prob)
- NumericVector [Rcpp::rnbinom\\_mu](#) (int n, double siz, double mu)
- NumericVector [Rcpp::rnchisq](#) (int n, double df, double lambda)
- NumericVector [Rcpp::rnchisq](#) (int n, double df)
- NumericVector [Rcpp::rpois](#) (int n, double mu)
- NumericVector [Rcpp::rsignrank](#) (int n, double nn)
- NumericVector [Rcpp::rt](#) (int n, double df)
- NumericVector [Rcpp::runif](#) (int n, double min, double max)
- NumericVector [Rcpp::runif](#) (int n, double min)
- NumericVector [Rcpp::runif](#) (int n)
- NumericVector [Rcpp::rweibull](#) (int n, double shape, double scale)
- NumericVector [Rcpp::rweibull](#) (int n, double shape)
- NumericVector [Rcpp::rwilcox](#) (int n, double mm, double nn)

## 7.200 inst/include/Rcpp/stats/random/rbeta.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::stats::BetaGenerator](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.201 inst/include/Rcpp/stats/random/rbinom.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

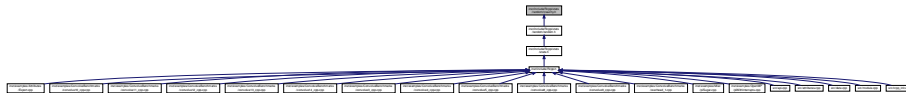
- class [Rcpp::stats::BinomGenerator](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.202 inst/include/Rcpp/stats/random/rcauchy.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::stats::CauchyGenerator](#)
- class [Rcpp::stats::CauchyGenerator\\_1](#)
- class [Rcpp::stats::CauchyGenerator\\_0](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.203 inst/include/Rcpp/stats/random/rchisq.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

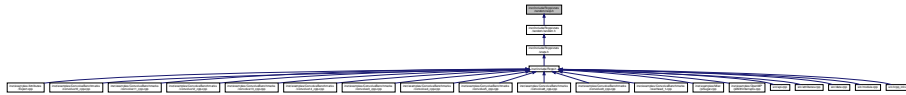
- class [Rcpp::stats::ChisqGenerator](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.204 inst/include/Rcpp/stats/random/rexp.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

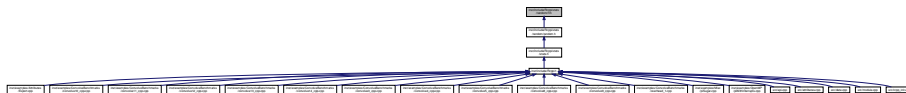
- class [Rcpp::stats::ExpGenerator](#)
- class [Rcpp::stats::ExpGenerator\\_\\_rate1](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.205 inst/include/Rcpp/stats/random/rf.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

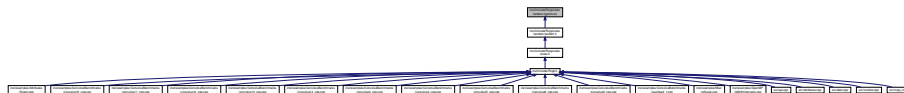
- class [Rcpp::stats::FGenerator\\_Finite\\_Finite](#)
- class [Rcpp::stats::FGenerator\\_NotFinite\\_Finite](#)
- class [Rcpp::stats::FGenerator\\_Finite\\_NotFinite](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.206 inst/include/Rcpp/stats/random/rgamma.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::stats::GammaGenerator](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.207 inst/include/Rcpp/stats/random/rgeom.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

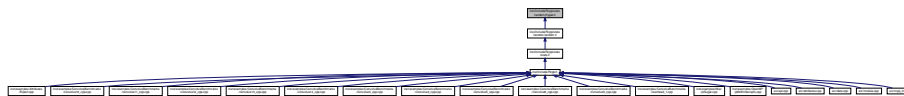
- class [Rcpp::stats::GeomGenerator](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.208 inst/include/Rcpp/stats/random/rhyper.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

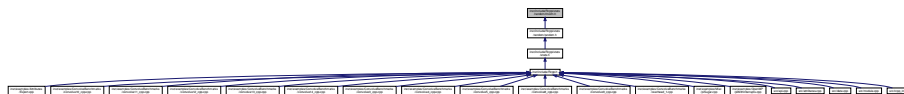
- class [Rcpp::stats::HyperGenerator](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.209 inst/include/Rcpp/stats/random/rlnorm.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

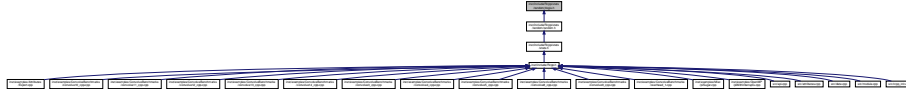
- class [Rcpp::stats::LNormGenerator](#)
- class [Rcpp::stats::LNormGenerator\\_1](#)
- class [Rcpp::stats::LNormGenerator\\_0](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.210 inst/include/Rcpp/stats/random/rlogis.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

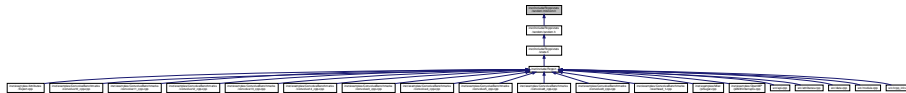
- class [Rcpp::stats::LogisGenerator](#)
- class [Rcpp::stats::LogisGenerator\\_1](#)
- class [Rcpp::stats::LogisGenerator\\_0](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.211 inst/include/Rcpp/stats/random/rnbinom.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::stats::NBinomGenerator](#)

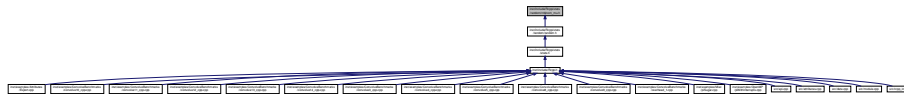
### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)



## 7.212 inst/include/Rcpp/stats/random/rnbinom\_mu.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

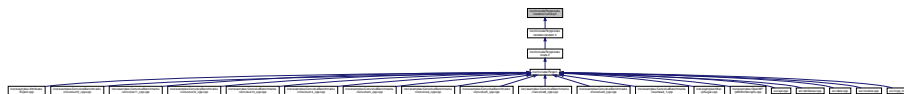
- class [Rcpp::stats::NBinomGenerator\\_Mu](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.213 inst/include/Rcpp/stats/random/rnchisq.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

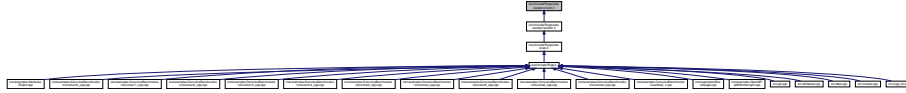
- class [Rcpp::stats::NChisqGenerator](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.214 inst/include/Rcpp/stats/random/rnorm.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::stats::NormGenerator](#)
- class [Rcpp::stats::NormGenerator\\_\\_sd1](#)
- class [Rcpp::stats::NormGenerator\\_\\_mean0](#)
- class [Rcpp::stats::NormGenerator\\_\\_mean0\\_\\_sd1](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.215 inst/include/Rcpp/stats/random/rpois.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

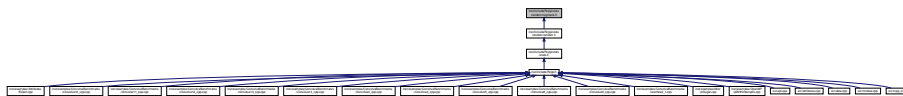
- class [Rcpp::stats::PoissonGenerator](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.216 inst/include/Rcpp/stats/random/rsignrank.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

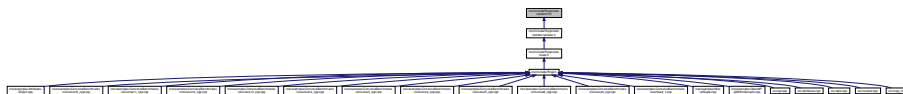
- class [Rcpp::stats::SignRankGenerator](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.217 inst/include/Rcpp/stats/random/rt.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

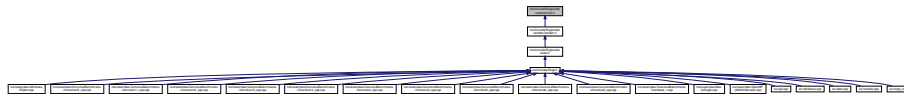
- class [Rcpp::stats::TGenerator](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.218 inst/include/Rcpp/stats/random/runif.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

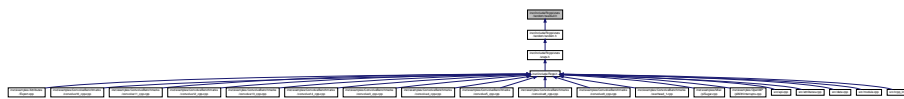
- class [Rcpp::stats::UnifGenerator](#)
- class [Rcpp::stats::UnifGenerator\\_\\_0\\_\\_1](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.219 inst/include/Rcpp/stats/random/rweibull.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

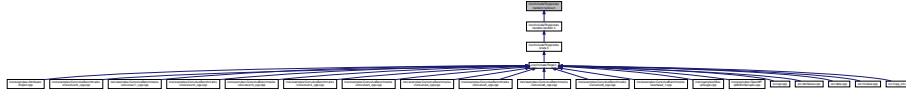
- class [Rcpp::stats::WeibullGenerator](#)
- class [Rcpp::stats::WeibullGenerator\\_\\_scale1](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.220 inst/include/Rcpp/stats/random/rwilcox.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::stats::WilcoxGenerator](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## 7.221 inst/include/Rcpp/stats/stats.h File Reference

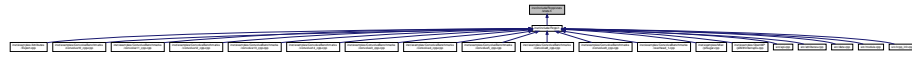
```
#include <Rcpp/stats/dpq/dpq.h>
#include <Rcpp/stats/unif.h>
#include <Rcpp/stats/norm.h>
#include <Rcpp/stats/gamma.h>
#include <Rcpp/stats/chisq.h>
#include <Rcpp/stats/beta.h>
#include <Rcpp/stats/t.h>
#include <Rcpp/stats/lnorm.h>
#include <Rcpp/stats/weibull.h>
#include <Rcpp/stats/logis.h>
#include <Rcpp/stats/f.h>
#include <Rcpp/stats/exp.h>
#include <Rcpp/stats/cauchy.h>
#include <Rcpp/stats/geom.h>
#include <Rcpp/stats/hyper.h>
#include <Rcpp/stats/nt.h>
#include <Rcpp/stats/nchisq.h>
#include <Rcpp/stats/nbeta.h>
#include <Rcpp/stats/nf.h>
#include <Rcpp/stats/nbinom.h>
#include <Rcpp/stats/nbinom_mu.h>
#include <Rcpp/stats/binom.h>
#include <Rcpp/stats/pois.h>
```

```
#include <Rcpp/stats/random/random.h>
```

Include dependency graph for stats.h:



This graph shows which files directly or indirectly include this file:



## Macros

- #define [ML\\_POSINF](#) R\_PosInf
- #define [ML\\_NEGINF](#) R\_NegInf
- #define [ML\\_NAN](#) R\_NaN

### 7.221.1 Macro Definition Documentation

#### 7.221.1.1 ML\_NAN

```
#define ML_NAN R_NaN
```

Definition at line 29 of file stats.h.

#### 7.221.1.2 ML\_NEGINF

```
#define ML_NEGINF R_NegInf
```

Definition at line 28 of file stats.h.

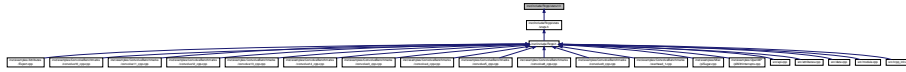
#### 7.221.1.3 ML\_POSINF

```
#define ML_POSINF R_PosInf
```

Definition at line 27 of file stats.h.

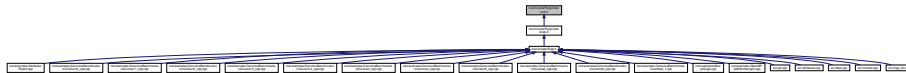
## 7.222 inst/include/Rcpp/stats/t.h File Reference

This graph shows which files directly or indirectly include this file:



## 7.223 inst/include/Rcpp/stats/unif.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

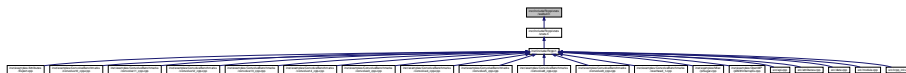
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

### Functions

- double [Rcpp::stats::dunif\\_1](#) (double x, double a, int [give\\_log](#))
- double [Rcpp::stats::dunif\\_0](#) (double x, int [give\\_log](#))
- double [Rcpp::stats::punif\\_1](#) (double x, double a, int [lower\\_tail](#), int [log\\_p](#))
- double [Rcpp::stats::punif\\_0](#) (double x, int [lower\\_tail](#), int [log\\_p](#))
- double [Rcpp::stats::qunif\\_1](#) (double p, double a, int [lower\\_tail](#), int [log\\_p](#))
- double [Rcpp::stats::qunif\\_0](#) (double p, int [lower\\_tail](#), int [log\\_p](#))

## 7.224 inst/include/Rcpp/stats/weibull.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

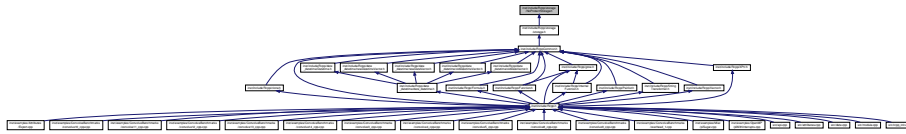
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::stats](#)

## Functions

- double [Rcpp::stats::dweibull\\_1](#) (double x, double shape, int [give\\_log](#))
- double [Rcpp::stats::pweibull\\_1](#) (double x, double shape, int lower\_tail, int log\_p)
- double [Rcpp::stats::qweibull\\_1](#) (double p, double shape, int lower\_tail, int log\_p)

## 7.225 inst/include/Rcpp/storage/NoProtectStorage.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::NoProtectStorage< CLASS >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.226 inst/include/Rcpp/storage/PreserveStorage.h File Reference

This graph shows which files directly or indirectly include this file:





## Classes

- class [Rcpp::PreserveStorage< CLASS >](#)

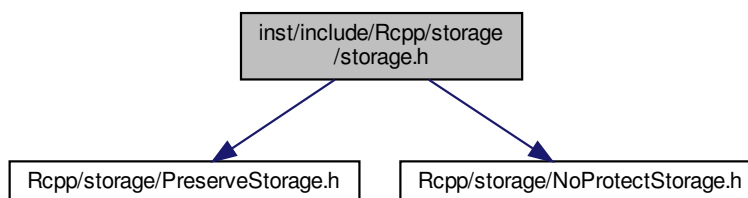
## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.227 inst/include/Rcpp/storage/storage.h File Reference

```
#include <Rcpp/storage/PreserveStorage.h>
#include <Rcpp/storage/NoProtectStorage.h>
```

Include dependency graph for storage.h:

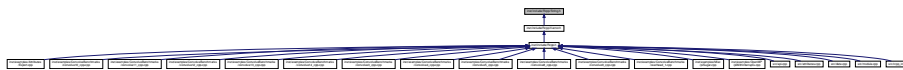


This graph shows which files directly or indirectly include this file:



## 7.228 inst/include/Rcpp/String.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::String](#)
- struct [Rcpp::traits::r\\_type\\_traits< Rcpp::String >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< Rcpp::String >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*
- [Rcpp::internal](#)  
*internal implementation details*

## Macros

- `#define RCPP_STRING_DEBUG_LEVEL 0`
- `#define RCPP_STRING_DEBUG(MSG)`
- `#define RCPP_STRING_DEBUG_1(fmt, MSG)`
- `#define RCPP_STRING_DEBUG_2(fmt, M1, M2)`
- `#define RCPP_STRING_DEBUG_3(fmt, M1, M2, M3)`

## Functions

- `template<> SEXP Rcpp::internal::make_charsexp< Rcpp::String > (const Rcpp::String &s)`
- `template<> SEXP Rcpp::wrap< Rcpp::String > (const Rcpp::String &object)`
- `bool Rcpp::operator== (const String::StringProxy &lhs, const String &rhs)`
- `bool Rcpp::operator!= (const String::StringProxy &lhs, const String &rhs)`
- `bool Rcpp::operator== (const String::const_StringProxy &lhs, const String &rhs)`
- `bool Rcpp::operator!= (const String::const_StringProxy &lhs, const String &rhs)`

### 7.228.1 Macro Definition Documentation

#### 7.228.1.1 RCPP\_STRING\_DEBUG

```
#define RCPP_STRING_DEBUG(  
    MSG )
```

Definition at line 37 of file String.h.

### 7.228.1.2 RCPP\_STRING\_DEBUG\_1

```
#define RCPP_STRING_DEBUG_1(
    fmt,
    MSG )
```

Definition at line 38 of file String.h.

### 7.228.1.3 RCPP\_STRING\_DEBUG\_2

```
#define RCPP_STRING_DEBUG_2(
    fmt,
    M1,
    M2 )
```

Definition at line 39 of file String.h.

### 7.228.1.4 RCPP\_STRING\_DEBUG\_3

```
#define RCPP_STRING_DEBUG_3(
    fmt,
    M1,
    M2,
    M3 )
```

Definition at line 40 of file String.h.

### 7.228.1.5 RCPP\_STRING\_DEBUG\_LEVEL

```
#define RCPP_STRING_DEBUG_LEVEL 0
```

Definition at line 26 of file String.h.

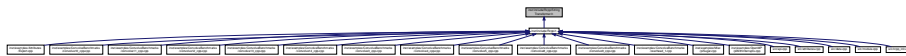
## 7.229 inst/include/Rcpp/StringTransformer.h File Reference

```
#include <RcppCommon.h>
```

Include dependency graph for StringTransformer.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::StringTransformer< UnaryOperator >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

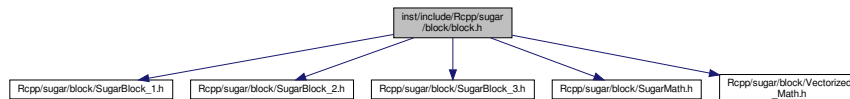
## Functions

- template<typename UnaryOperator >  
StringTransformer< UnaryOperator > [Rcpp::make\\_string\\_transformer](#) (const UnaryOperator &fun)

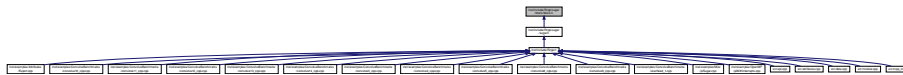
## 7.230 inst/include/Rcpp/sugar/block/block.h File Reference

```
#include <Rcpp/sugar/block/SugarBlock_1.h>
#include <Rcpp/sugar/block/SugarBlock_2.h>
#include <Rcpp/sugar/block/SugarBlock_3.h>
#include <Rcpp/sugar/block/SugarMath.h>
#include <Rcpp/sugar/block/Vectorized_Math.h>
```

Include dependency graph for block.h:



This graph shows which files directly or indirectly include this file:



## 7.231 inst/include/Rcpp/sugar/block/SugarBlock\_1.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::SugarBlock\\_1](#)< NA, RESULT\_TYPE, U1, T1 >

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Macros

- `#define SB1_T VectorBase<REALSXP,NA,T>`
- `#define SUGAR_BLOCK_1(__NAME__, __SYMBOL__)`

### 7.231.1 Macro Definition Documentation

#### 7.231.1.1 SB1\_T

```
#define SB1_T VectorBase<REALSXP,NA,T>
```

Definition at line 47 of file SugarBlock\_1.h.

#### 7.231.1.2 SUGAR\_BLOCK\_1

```
#define SUGAR_BLOCK_1(  
    __NAME__,  
    __SYMBOL__ )
```

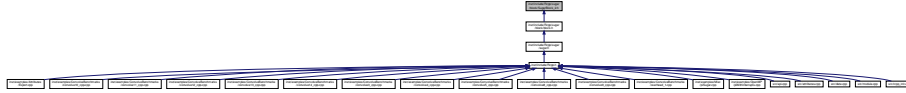
#### Value:

```
namespace Rcpp{  
template <bool NA, typename T>  
inline sugar::SugarBlock_1<NA, double, double, SB1_T >  
__NAME__(  
    const SB1_T& t  
) {  
    return sugar::SugarBlock_1<NA, double, double, SB1_T >(  
        __SYMBOL__ , t  
    ) ;  
}  
}
```

Definition at line 49 of file SugarBlock\_1.h.

## 7.232 inst/include/Rcpp/sugar/block/SugarBlock\_2.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::SugarBlock\\_2< NA, RESULT\\_TYPE, U1, T1, U2, T2 >](#)
- class [Rcpp::sugar::SugarBlock\\_2\\_\\_VP< NA, RESULT\\_TYPE, U1, T1, U2 >](#)
- class [Rcpp::sugar::SugarBlock\\_2\\_\\_PV< NA, RESULT\\_TYPE, U1, U2, T2 >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Macros

- `#define SB2_LHT VectorBase<REALSXP,LHS_NA,LHS_T>`
- `#define SB2_RHT VectorBase<REALSXP,RHS_NA,RHS_T>`
- `#define SUGAR_BLOCK_2(__NAME__, __SYMBOL__)`
- `#define SUGAR_BLOCK_2_NA(__NAME__, __SYMBOL__, __NA__)`

### 7.232.1 Macro Definition Documentation

#### 7.232.1.1 SB2\_LHT

```
#define SB2_LHT VectorBase<REALSXP,LHS_NA,LHS_T>
```

Definition at line 89 of file SugarBlock\_2.h.

### 7.232.1.2 SB2\_RHT

```
#define SB2_RHT VectorBase<REALSXP, RHS_NA, RHS_T>
```

Definition at line 90 of file SugarBlock\_2.h.

### 7.232.1.3 SUGAR\_BLOCK\_2

```
#define SUGAR_BLOCK_2(  
    __NAME__,  
    __SYMBOL__ )
```

Definition at line 92 of file SugarBlock\_2.h.

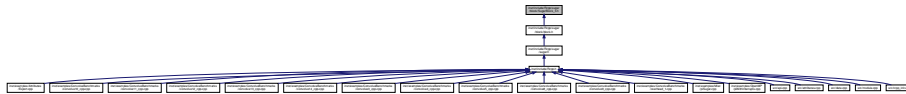
### 7.232.1.4 SUGAR\_BLOCK\_2\_NA

```
#define SUGAR_BLOCK_2_NA(  
    __NAME__,  
    __SYMBOL__,  
    __NA__ )
```

Definition at line 127 of file SugarBlock\_2.h.

## 7.233 inst/include/Rcpp/sugar/block/SugarBlock\_3.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::SugarBlock\\_3\\_VVV< NA, RESULT\\_TYPE, U1, T1, U2, T2, U3, T3 >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Macros

- #define [SB3\\_T1](#) VectorBase<REALSXP,T1\_NA,T1>
- #define [SB3\\_T2](#) VectorBase<REALSXP,T2\_NA,T2>
- #define [SB3\\_T3](#) VectorBase<REALSXP,T3\_NA,T3>
- #define [SUGAR\\_BLOCK\\_3](#)(\_\_NAME\_\_, \_\_SYMBOL\_\_)

### 7.233.1 Macro Definition Documentation

#### 7.233.1.1 SB3\_T1

```
#define SB3_T1 VectorBase<REALSXP,T1_NA,T1>
```

Definition at line 97 of file SugarBlock\_3.h.

#### 7.233.1.2 SB3\_T2

```
#define SB3_T2 VectorBase<REALSXP,T2_NA,T2>
```

Definition at line 98 of file SugarBlock\_3.h.

#### 7.233.1.3 SB3\_T3

```
#define SB3_T3 VectorBase<REALSXP,T3_NA,T3>
```

Definition at line 99 of file SugarBlock\_3.h.



## 7.233.1.4 SUGAR\_BLOCK\_3

```
#define SUGAR_BLOCK_3(
    __NAME__,
    __SYMBOL__ )
```

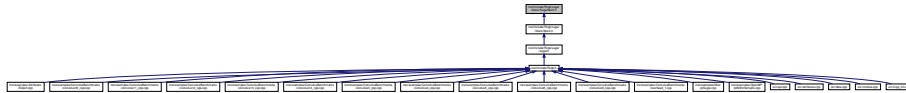
**Value:**

```
namespace Rcpp{
template <bool T1_NA, typename T1, bool T2_NA, typename T2, bool T3_NA, typename T3>
inline sugar::SugarBlock_3_VVV<
    (T1_NA||T2_NA||T3_NA) , double,
    double, SB3_T1,
    double, SB3_T2,
    double, SB3_T3
>
__NAME__(
    const SB3_T1& x1,
    const SB3_T2& x2,
    const SB3_T3& x3
){
    return sugar::SugarBlock_3_VVV<
        (T1_NA||T2_NA||T3_NA) , double,
        double, SB3_T1,
        double, SB3_T2,
        double, SB3_T3
    >(
        __SYMBOL__ , x1, x2, x3
    ) ;
}
}
```

Definition at line 101 of file SugarBlock\_3.h.

## 7.234 inst/include/Rcpp/sugar/block/SugarMath.h File Reference

This graph shows which files directly or indirectly include this file:

**Classes**

- class [Rcpp::sugar::SugarMath\\_1](#) < NA, RESULT\_TYPE, U1, T1, FunPtr >
- class [Rcpp::sugar::SugarMath\\_1](#) < NA, RESULT\_TYPE, int, T1, FunPtr >
- class [Rcpp::sugar::SugarMath\\_1](#) < false, RESULT\_TYPE, int, T1, FunPtr >

**Namespaces**

- [Rcpp](#)
  - *Rcpp API.*
- [Rcpp::sugar](#)

## Macros

- #define [SUGAR\\_MATH\\_1](#)(\_\_NAME\_\_, \_\_SYMBOL\_\_)

### 7.234.1 Macro Definition Documentation

#### 7.234.1.1 SUGAR\_MATH\_1

```
#define SUGAR_MATH_1(
    __NAME__,
    __SYMBOL__ )
```

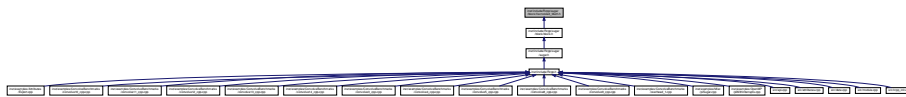
#### Value:

```
namespace Rcpp{
template <bool NA, typename T>
inline sugar::SugarMath_1<NA,double,double,T, double (*) (double) >
__NAME__(
    const VectorBase<REALSXP,NA,T>& t
){
    return sugar::SugarMath_1<NA,double,double,T, double (*) (double)>(
        &__SYMBOL__ , t
    ) ;
}
inline sugar::SugarMath_1<true,double,double,NumericVector, double(*) (double)>
__NAME__( SEXP x){ return __NAME__( NumericVector( x ) ) ; }
template <bool NA, typename T>
inline sugar::SugarMath_1<NA,double,int,T, double (*) (double) >
__NAME__(
    const VectorBase<INTSXP,NA,T>& t
){
    return sugar::SugarMath_1<NA,double,int,T, double (*) (double)>(
        &__SYMBOL__ , t
    ) ;
}
}
```

Definition at line 100 of file SugarMath.h.

## 7.235 inst/include/Rcpp/sugar/block/Vectorized\_Math.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Vectorized](#)< Func, NA, VEC >
- class [Rcpp::sugar::Vectorized\\_INTSXP](#)< Func, NA, VEC >
- class [Rcpp::sugar::Vectorized\\_INTSXP](#)< Func, false, VEC >

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Macros

- #define [VECTORIZED\\_MATH\\_1](#)(\_\_NAME\_\_, \_\_SYMBOL\_\_)

## Typedefs

- typedef double(\* [Rcpp::sugar::DDFun](#)) (double)

## 7.235.1 Macro Definition Documentation

### 7.235.1.1 VECTORIZED\_MATH\_1

```
#define VECTORIZED_MATH_1(
    __NAME__,
    __SYMBOL__ )
```

#### Value:

```
namespace Rcpp{
  template <bool NA, typename T>
  inline sugar::Vectorized<__SYMBOL__,NA,T>
  __NAME__( const VectorBase<REALSXP,NA,T>& t ){
    return sugar::Vectorized<__SYMBOL__,NA,T>( t ) ;
  }
  inline sugar::Vectorized<__SYMBOL__,true,NumericVector>
  __NAME__( SEXP x){ return __NAME__( NumericVector( x ) ) ; }
  template <bool NA, typename T>
  inline sugar::Vectorized<INTSXP<__SYMBOL__,NA,T>
  __NAME__( const VectorBase<INTSXP,NA,T>& t
    ) {
    return sugar::Vectorized<INTSXP<__SYMBOL__,NA,T>( t ) ;
  }
}
```

Definition at line 83 of file Vectorized\_Math.h.

## 7.236 inst/include/Rcpp/sugar/functions/all.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::All< NA, T >](#)
- class [Rcpp::sugar::All< false, T >](#)

## Namespaces

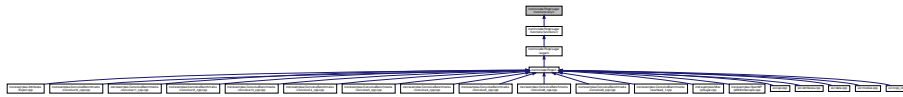
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<bool NA, typename T >`  
`sugar::All< NA, T >` [Rcpp::all](#) (const [Rcpp::VectorBase](#)< LGLSXP, NA, T > &t)

## 7.237 inst/include/Rcpp/sugar/functions/any.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Any< NA, T >](#)
- class [Rcpp::sugar::Any< false, T >](#)

## Namespaces

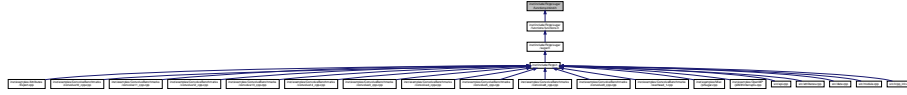
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<bool NA, typename T >`  
`sugar::Any< NA, T >` [Rcpp::any](#) (const [Rcpp::VectorBase](#)< LGLSXP, NA, T > &t)

## 7.238 inst/include/Rcpp/sugar/functions/cbind.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::sugar::cbind\\_impl::cbind\\_sexptype\\_traits< T >](#)
- struct [Rcpp::sugar::cbind\\_impl::cbind\\_sexptype\\_traits< SEXP >](#)
- struct [Rcpp::sugar::cbind\\_impl::cbind\\_storage\\_type< RTYPE >](#)
- struct [Rcpp::sugar::cbind\\_impl::cbind\\_storage\\_type< LGLSXP >](#)
- class [Rcpp::sugar::cbind\\_impl::BindableExpression< RTYPE, E >](#)
- class [Rcpp::sugar::cbind\\_impl::ContainerBindable< RTYPE, T >](#)
- struct [Rcpp::sugar::cbind\\_impl::scalar< RTYPE >](#)
- class [Rcpp::sugar::cbind\\_impl::ScalarBindable< T >](#)
- class [Rcpp::sugar::cbind\\_impl::JoinOp< RTYPE, E1, E2 >](#)
- class [Rcpp::sugar::cbind\\_impl::JoinOp< RTYPE, E1, ScalarBindable< typename scalar< RTYPE >::type > >](#)
- class [Rcpp::sugar::cbind\\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, E2 >](#)
- class [Rcpp::sugar::cbind\\_impl::JoinOp< RTYPE, ScalarBindable< typename scalar< RTYPE >::type >, ScalarBindable< typena](#)
- class [Rcpp::sugar::cbind\\_impl::detail::has\\_stored\\_type< T >](#)
- struct [Rcpp::sugar::cbind\\_impl::detail::has\\_stored\\_type< T >::no](#)
- struct [Rcpp::sugar::cbind\\_impl::detail::MakeBindableCall< T, is\\_container >](#)
- struct [Rcpp::sugar::cbind\\_impl::detail::MakeBindableCall< T, true >](#)
- struct [Rcpp::sugar::cbind\\_impl::detail::MakeBindableCall< Rcpp::Matrix< LGLSXP >, true >](#)
- struct [Rcpp::sugar::cbind\\_impl::detail::MakeBindableCall< Rcpp::Vector< LGLSXP >, true >](#)
- struct [Rcpp::sugar::cbind\\_impl::detail::MakeBindableCall< T, false >](#)
- struct [Rcpp::sugar::cbind\\_impl::detail::matrix\\_return< T, is\\_container >](#)
- struct [Rcpp::sugar::cbind\\_impl::detail::matrix\\_return< T, true >](#)
- struct [Rcpp::sugar::cbind\\_impl::detail::matrix\\_return< Rcpp::Matrix< LGLSXP >, true >](#)
- struct [Rcpp::sugar::cbind\\_impl::detail::matrix\\_return< Rcpp::Vector< LGLSXP >, true >](#)
- struct [Rcpp::sugar::cbind\\_impl::detail::matrix\\_return< bool, false >](#)
- struct [Rcpp::sugar::cbind\\_impl::detail::matrix\\_return< T, false >](#)
- struct [Rcpp::sugar::cbind\\_impl::matrix\\_return< T, B >](#)
- struct [Rcpp::sugar::cbind\\_impl::matrix\\_return< T, false >](#)

### Namespaces

- [Rcpp](#)
  - *Rcpp API.*
- [Rcpp::sugar](#)
- [Rcpp::sugar::cbind\\_impl](#)
- [Rcpp::sugar::cbind\\_impl::detail](#)

## Macros

- `#define MakeBindable\(x\) (cbind_impl::detail::MakeBindable(x)(x))`

## Functions

- `template<int RTYPE>`  
`ContainerBindable< RTYPE, Rcpp::Matrix< RTYPE > > Rcpp::sugar::cbind\_impl::MakeContainerBindable`  
`(const Rcpp::Matrix< RTYPE > &x)`
- `template<int RTYPE>`  
`ContainerBindable< RTYPE, Rcpp::Vector< RTYPE > > Rcpp::sugar::cbind\_impl::MakeContainerBindable`  
`(const Rcpp::Vector< RTYPE > &x)`
- `template<>` `ContainerBindable< LGLSXP, Rcpp::Matrix< LGLSXP > > Rcpp::sugar::cbind\_impl::MakeContainerBindable`  
`(const Rcpp::Matrix< LGLSXP > &x)`
- `template<>` `ContainerBindable< LGLSXP, Rcpp::Vector< LGLSXP > > Rcpp::sugar::cbind\_impl::MakeContainerBindable`  
`(const Rcpp::Vector< LGLSXP > &x)`
- `template<typename T >`  
`ScalarBindable< T > Rcpp::sugar::cbind\_impl::MakeScalarBindable (const T &t)`
- `template<int RTYPE, typename E1 , typename E2 >`  
`JoinOp< RTYPE, E1, E2 > Rcpp::sugar::cbind\_impl::operator, (const BindableExpression< RTYPE, E1 > &e1,`  
`const BindableExpression< RTYPE, E2 > &e2)`
- `template<typename T >`  
`Rcpp::traits::enable\_if< has\_stored\_type< T >::value, MakeBindableCall< T, true > >::type Rcpp::sugar::cbind\\_impl::detail::Make`  
`(const T &t)`
- `template<typename T >`  
`Rcpp::traits::enable\_if< !has\_stored\_type< T >::value, MakeBindableCall< T, false > >::type Rcpp::sugar::cbind\\_impl::detail::Make`  
`(const T &t)`
- `template<typename T1 , typename T2 >`  
`cbind_impl::matrix_return< T1 >::type Rcpp::sugar::cbind (const T1 &t1, const T2 &t2)`
- `template<typename T1 , typename T2 , typename T3 >`  
`cbind_impl::matrix_return< T1 >::type Rcpp::sugar::cbind (const T1 &t1, const T2 &t2, const T3 &t3)`
- `template<typename T1 , typename T2 , typename T3 , typename T4 >`  
`cbind_impl::matrix_return< T1 >::type Rcpp::sugar::cbind (const T1 &t1, const T2 &t2, const T3 &t3, const T4`  
`&t4)`
- `template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 >`  
`cbind_impl::matrix_return< T1 >::type Rcpp::sugar::cbind (const T1 &t1, const T2 &t2, const T3 &t3, const T4`  
`&t4, const T5 &t5)`
- `template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 >`  
`cbind_impl::matrix_return< T1 >::type Rcpp::sugar::cbind (const T1 &t1, const T2 &t2, const T3 &t3, const T4`  
`&t4, const T5 &t5, const T6 &t6)`
- `template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 >`  
`cbind_impl::matrix_return< T1 >::type Rcpp::sugar::cbind (const T1 &t1, const T2 &t2, const T3 &t3, const T4`  
`&t4, const T5 &t5, const T6 &t6, const T7 &t7)`
- `template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 >`  
`cbind_impl::matrix_return< T1 >::type Rcpp::sugar::cbind (const T1 &t1, const T2 &t2, const T3 &t3, const T4`  
`&t4, const T5 &t5, const T6 &t6, const T7 &t7, const T8 &t8)`
- `template<typename T1 , typename T2 , typename T3 , typename T4 , typename T5 , typename T6 , typename T7 , typename T8 , typename`  
`T9 >`  
`cbind_impl::matrix_return< T1 >::type Rcpp::sugar::cbind (const T1 &t1, const T2 &t2, const T3 &t3, const T4`  
`&t4, const T5 &t5, const T6 &t6, const T7 &t7, const T8 &t8, const T9 &t9)`

















```
const T39 &t39, const T40 &t40, const T41 &t41, const T42 &t42, const T43 &t43, const T44 &t44, const T45
&t45, const T46 &t46, const T47 &t47, const T48 &t48, const T49 &t49, const T50 &t50)
```

## 7.238.1 Macro Definition Documentation

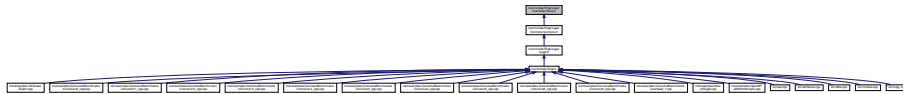
### 7.238.1.1 MakeBindable

```
#define MakeBindable(
    x ) (cbind_impl::detail::MakeBindable(x)(x))
```

Definition at line 510 of file cbind.h.

## 7.239 inst/include/Rcpp/sugar/functions/clamp.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::sugar::clamp\\_operator< RTYPE, NA >](#)
- struct [Rcpp::sugar::clamp\\_operator< REALSXP, true >](#)
- class [Rcpp::sugar::Clamp\\_Primitive\\_Vector\\_Primitive< RTYPE, NA, T >](#)

### Namespaces

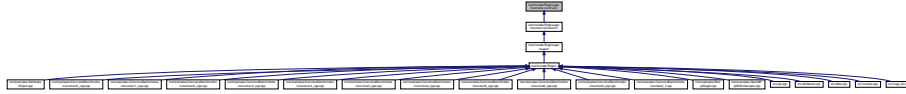
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- `template<int RTYPE, bool NA, typename T >`  
[sugar::Clamp\\_Primitive\\_Vector\\_Primitive< RTYPE, NA, T >](#) [Rcpp::clamp](#) (typename [Rcpp::traits::storage\\_type<](#)  
 RTYPE >::type lhs, const [Rcpp::VectorBase< RTYPE, NA, T >](#) &vec, typename [Rcpp::traits::storage\\_type<](#)  
 RTYPE >::type rhs)

## 7.240 inst/include/Rcpp/sugar/functions/cummax.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Cummax< RTYPE, NA, T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- `template<bool NA, typename T >`  
`sugar::Cummax< INTSXP, NA, T >` [Rcpp::cummax](#) (const VectorBase< INTSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Cummax< REALSXP, NA, T >` [Rcpp::cummax](#) (const VectorBase< REALSXP, NA, T > &t)

## 7.241 inst/include/Rcpp/sugar/functions/cummin.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Cummin< RTYPE, NA, T >](#)

### Namespaces

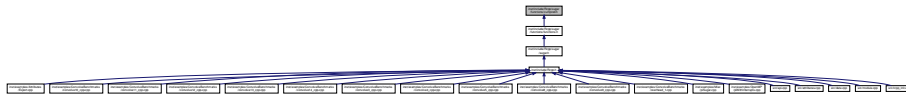
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<bool NA, typename T >`  
`sugar::Cummin< INTSXP, NA, T >` [Rcpp::cummin](#) (const VectorBase< INTSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Cummin< REALSXP, NA, T >` [Rcpp::cummin](#) (const VectorBase< REALSXP, NA, T > &t)

## 7.242 inst/include/Rcpp/sugar/functions/cumprod.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Cumprod< RTYPE, NA, T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<bool NA, typename T >`  
`sugar::Cumprod< INTSXP, NA, T >` [Rcpp::cumprod](#) (const VectorBase< INTSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Cumprod< REALSXP, NA, T >` [Rcpp::cumprod](#) (const VectorBase< REALSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Cumprod< CPLXSXP, NA, T >` [Rcpp::cumprod](#) (const VectorBase< CPLXSXP, NA, T > &t)

## 7.243 inst/include/Rcpp/sugar/functions/cumsum.h File Reference

This graph shows which files directly or indirectly include this file:





## Classes

- class [Rcpp::sugar::Cumsum< RTYPE, NA, T >](#)

## Namespaces

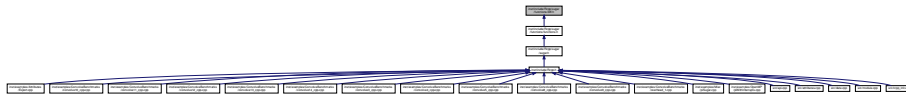
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<bool NA, typename T >`  
`sugar::Cumsum< INTSXP, NA, T >` [Rcpp::cumsum](#) (`const VectorBase< INTSXP, NA, T > &t`)
- `template<bool NA, typename T >`  
`sugar::Cumsum< REALSXP, NA, T >` [Rcpp::cumsum](#) (`const VectorBase< REALSXP, NA, T > &t`)

## 7.244 inst/include/Rcpp/sugar/functions/diff.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Diff< RTYPE, LHS\\_NA, LHS\\_T >](#)
- class [Rcpp::sugar::Diff< REALSXP, LHS\\_NA, LHS\\_T >](#)
- class [Rcpp::sugar::Diff< RTYPE, false, LHS\\_T >](#)

## Namespaces

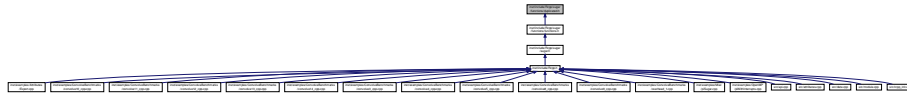
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<bool LHS_NA, typename LHS_T >`  
`sugar::Diff< INTSXP, LHS_NA, LHS_T >` [Rcpp::diff](#) (`const VectorBase< INTSXP, LHS_NA, LHS_T > &lhs`)
- `template<bool LHS_NA, typename LHS_T >`  
`sugar::Diff< REALSXP, LHS_NA, LHS_T >` [Rcpp::diff](#) (`const VectorBase< REALSXP, LHS_NA, LHS_T > &lhs`)

## 7.245 inst/include/Rcpp/sugar/functions/duplicated.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [Rcpp](#)  
*Rcpp API.*

### Functions

- `template<int RTYPE, bool NA, typename T >`  
LogicalVector [Rcpp::duplicated](#) (const VectorBase< RTYPE, NA, T > &x)

## 7.246 inst/include/Rcpp/sugar/functions/functions.h File Reference

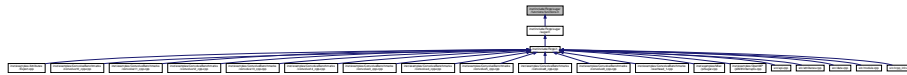
```
#include <cmath>
#include <Rcpp/sugar/functions/Lazy.h>
#include <Rcpp/sugar/functions/math.h>
#include <Rcpp/sugar/functions/complex.h>
#include <Rcpp/sugar/functions/any.h>
#include <Rcpp/sugar/functions/all.h>
#include <Rcpp/sugar/functions/is_na.h>
#include <Rcpp/sugar/functions/is_finite.h>
#include <Rcpp/sugar/functions/is_infinite.h>
#include <Rcpp/sugar/functions/is_nan.h>
#include <Rcpp/sugar/functions/na_omit.h>
#include <Rcpp/sugar/functions/seq_along.h>
#include <Rcpp/sugar/functions/sapply.h>
#include <Rcpp/sugar/functions/mapply.h>
#include <Rcpp/sugar/functions/lapply.h>
#include <Rcpp/sugar/functions/ifelse.h>
#include <Rcpp/sugar/functions/pmin.h>
#include <Rcpp/sugar/functions/pmax.h>
#include <Rcpp/sugar/functions/clamp.h>
#include <Rcpp/sugar/functions/min.h>
#include <Rcpp/sugar/functions/max.h>
#include <Rcpp/sugar/functions/range.h>
#include <Rcpp/sugar/functions/sign.h>
#include <Rcpp/sugar/functions/diff.h>
#include <Rcpp/sugar/functions/pow.h>
#include <Rcpp/sugar/functions/rep.h>
```

```

#include <Rcpp/sugar/functions/rep_len.h>
#include <Rcpp/sugar/functions/rep_each.h>
#include <Rcpp/sugar/functions/rev.h>
#include <Rcpp/sugar/functions/head.h>
#include <Rcpp/sugar/functions/tail.h>
#include <Rcpp/sugar/functions/sum.h>
#include <Rcpp/sugar/functions/mean.h>
#include <Rcpp/sugar/functions/var.h>
#include <Rcpp/sugar/functions/sd.h>
#include <Rcpp/sugar/functions/cumsum.h>
#include <Rcpp/sugar/functions/which_min.h>
#include <Rcpp/sugar/functions/which_max.h>
#include <Rcpp/sugar/functions/unique.h>
#include <Rcpp/sugar/functions/match.h>
#include <Rcpp/sugar/functions/table.h>
#include <Rcpp/sugar/functions/duplicated.h>
#include <Rcpp/sugar/functions/self_match.h>
#include <Rcpp/sugar/functions/setdiff.h>
#include <Rcpp/sugar/functions/strings/strings.h>
#include <Rcpp/sugar/functions/cumprod.h>
#include <Rcpp/sugar/functions/cummin.h>
#include <Rcpp/sugar/functions/cummax.h>
#include <Rcpp/sugar/functions/median.h>
#include <Rcpp/sugar/functions/cbind.h>
#include <Rcpp/sugar/functions/rowSums.h>
#include <Rcpp/sugar/functions/sample.h>

```

This graph shows which files directly or indirectly include this file:



## 7.247 inst/include/Rcpp/sugar/functions/head.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Head< RTYPE, NA, T >](#)

### Namespaces

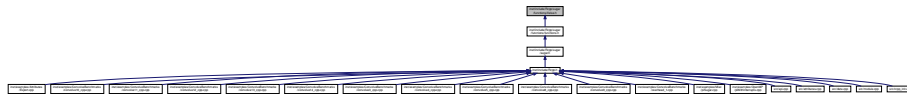
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::Head< RTYPE, NA, T >` [Rcpp::head](#) (const VectorBase< RTYPE, NA, T > &t, R\_xlen\_t n)

## 7.248 inst/include/Rcpp/sugar/functions/ifelse.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::IfElse< RTYPE, COND\\_NA, COND\\_T, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::IfElse< RTYPE, false, COND\\_T, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::IfElse\\_Primitive\\_Vector< RTYPE, COND\\_NA, COND\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::IfElse\\_Primitive\\_Vector< RTYPE, false, COND\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::IfElse\\_Vector\\_Primitive< RTYPE, COND\\_NA, COND\\_T, LHS\\_NA, LHS\\_T >](#)
- class [Rcpp::sugar::IfElse\\_Vector\\_Primitive< RTYPE, false, COND\\_T, LHS\\_NA, LHS\\_T >](#)
- class [Rcpp::sugar::IfElse\\_Primitive\\_Primitive< RTYPE, COND\\_NA, COND\\_T >](#)
- class [Rcpp::sugar::IfElse\\_Primitive\\_Primitive< RTYPE, false, COND\\_T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

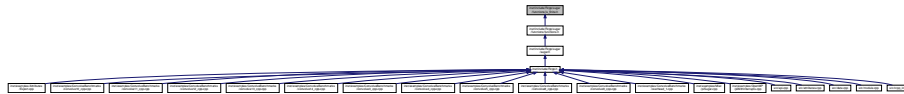
## Functions

- `template<int RTYPE, bool COND_NA, typename COND_T, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`sugar::IfElse< RTYPE, COND_NA, COND_T, LHS_NA, LHS_T, RHS_NA, RHS_T >` [Rcpp::ifelse](#) (const [Rcpp::VectorBase](#)< LGLSXP, COND\_NA, COND\_T > &cond, const [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, const [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)
- `template<int RTYPE, bool COND_NA, typename COND_T, bool RHS_NA, typename RHS_T >`  
`sugar::IfElse_Primitive_Vector< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >` [Rcpp::ifelse](#) (const [Rcpp::VectorBase](#)< LGLSXP, COND\_NA, COND\_T > &cond, typename traits::storage\_type< RTYPE >::type lhs, const [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)
- `template<int RTYPE, bool COND_NA, typename COND_T, bool RHS_NA, typename RHS_T >`  
`sugar::IfElse_Vector_Primitive< RTYPE, COND_NA, COND_T, RHS_NA, RHS_T >` [Rcpp::ifelse](#) (const [Rcpp::VectorBase](#)< LGLSXP, COND\_NA, COND\_T > &cond, const [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &lhs, typename traits::storage\_type< RTYPE >::type rhs)

- `template<bool COND_NA, typename COND_T >`  
`sugar::IfElse_Primitive_Primitive< REALSXP, COND_NA, COND_T >` [Rcpp::ifelse](#) (const [Rcpp::VectorBase](#)< LGLSXP, COND\_NA, COND\_T > &cond, double lhs, double rhs)
- `template<bool COND_NA, typename COND_T >`  
`sugar::IfElse_Primitive_Primitive< INTSXP, COND_NA, COND_T >` [Rcpp::ifelse](#) (const [Rcpp::VectorBase](#)< LGLSXP, COND\_NA, COND\_T > &cond, int lhs, int rhs)
- `template<bool COND_NA, typename COND_T >`  
`sugar::IfElse_Primitive_Primitive< CPLXSXP, COND_NA, COND_T >` [Rcpp::ifelse](#) (const [Rcpp::VectorBase](#)< LGLSXP, COND\_NA, COND\_T > &cond, Rcomplex lhs, Rcomplex rhs)
- `template<bool COND_NA, typename COND_T >`  
`sugar::IfElse_Primitive_Primitive< LGLSXP, COND_NA, COND_T >` [Rcpp::ifelse](#) (const [Rcpp::VectorBase](#)< LGLSXP, COND\_NA, COND\_T > &cond, bool lhs, bool rhs)

## 7.249 inst/include/Rcpp/sugar/functions/is\_finite.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::IsFinite](#)< RTYPE, NA, VEC\_TYPE >

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::IsFinite`< RTYPE, NA, T > [Rcpp::is\\_finite](#) (const [Rcpp::VectorBase](#)< RTYPE, NA, T > &t)

## 7.250 inst/include/Rcpp/traits/is\_finite.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

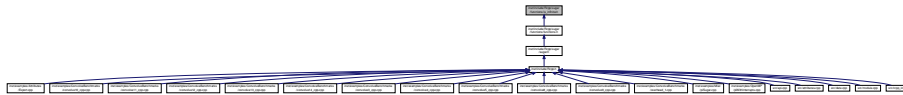
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## Functions

- `template<int RTYPE>`  
`bool Rcpp::traits::is\_finite (typename storage_type< RTYPE >::type)`
- `template<> bool Rcpp::traits::is\_finite< INTSXP > (int x)`
- `template<> bool Rcpp::traits::is\_finite< REALSXP > (double x)`
- `template<> bool Rcpp::traits::is\_finite< CPLXSXP > (Rcomplex x)`
- `template<> bool Rcpp::traits::is\_finite< STRSXP > (SEXP)`
- `template<> bool Rcpp::traits::is\_finite< LGLSXP > (int x)`

## 7.251 inst/include/Rcpp/sugar/functions/is\_infinite.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::IsInfinite](#)< RTYPE, NA, VEC\_TYPE >

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::IsInfinite< RTYPE, NA, T > Rcpp::is\_infinite (const Rcpp::VectorBase< RTYPE, NA, T > &t)`

## 7.252 inst/include/Rcpp/traits/is\_infinite.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

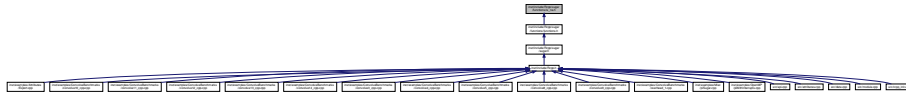
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

### Functions

- `template<int RTYPE>`  
`bool Rcpp::traits::is\_infinite (typename storage_type< RTYPE >::type)`
- `template<> bool Rcpp::traits::is\_infinite< REALSXP > (double x)`
- `template<> bool Rcpp::traits::is\_infinite< CPLXSXP > (Rcomplex x)`

## 7.253 inst/include/Rcpp/sugar/functions/is\_na.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::IsNa](#)< RTYPE, NA, VEC\_TYPE >
- class [Rcpp::sugar::IsNa](#)< RTYPE, false, VEC\_TYPE >
- class [Rcpp::sugar::IsNa\\_Vector\\_is\\_na](#)< T >

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::IsNa< RTYPE, NA, T > Rcpp::is_na (const Rcpp::VectorBase< RTYPE, NA, T > &t)`
- `sugar::IsNa_Vector_is_na< oldDatetimeVector > Rcpp::is_na (const oldDatetimeVector &x)`
- `sugar::IsNa_Vector_is_na< oldDateVector > Rcpp::is_na (const oldDateVector &x)`
- `sugar::IsNa_Vector_is_na< NumericVector > Rcpp::is_na (newDatetimeVector &x)`
- `sugar::IsNa_Vector_is_na< NumericVector > Rcpp::is_na (newDateVector &x)`

## 7.254 inst/include/Rcpp/traits/is\_na.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

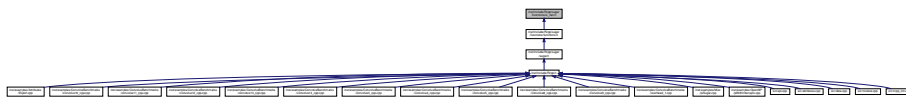
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## Functions

- `template<int RTYPE>`  
`bool Rcpp::traits::is_na (typename storage_type< RTYPE >::type)`
- `template<> bool Rcpp::traits::is_na< INTSXP > (int x)`
- `template<> bool Rcpp::traits::is_na< REALSXP > (double x)`
- `template<> bool Rcpp::traits::is_na< CPLXSXP > (Rcomplex x)`
- `template<> bool Rcpp::traits::is_na< STRSXP > (SEXP x)`
- `template<> bool Rcpp::traits::is_na< LGLSXP > (int x)`

## 7.255 inst/include/Rcpp/sugar/functions/is\_nan.h File Reference

This graph shows which files directly or indirectly include this file:





## Classes

- class [Rcpp::sugar::IsNaN< RTYPE, NA, VEC\\_TYPE >](#)

## Namespaces

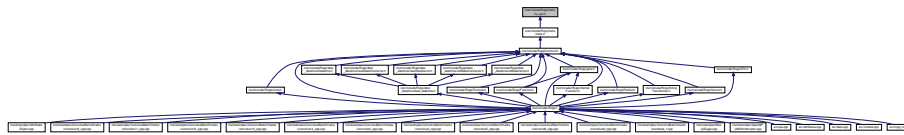
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::IsNaN< RTYPE, NA, T >` [Rcpp::is\\_nan](#) (const [Rcpp::VectorBase](#)< RTYPE, NA, T > &t)

## 7.256 inst/include/Rcpp/traits/is\_nan.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

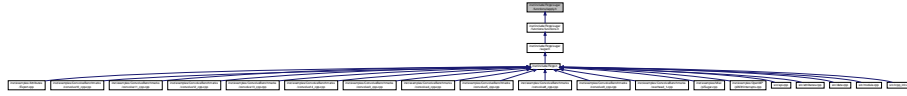
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## Functions

- `template<int RTYPE>`  
`bool` [Rcpp::traits::is\\_nan](#) (typename `storage_type`< RTYPE >::type)
- `template<>` `bool` [Rcpp::traits::is\\_nan](#)< [REALSXP](#) > (double x)
- `template<>` `bool` [Rcpp::traits::is\\_nan](#)< [CPLXSXP](#) > (Rcomplex x)

## 7.257 inst/include/Rcpp/sugar/functions/lapply.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Lapply< RTYPE, NA, T, Function >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- `template<int RTYPE, bool NA, typename T , typename Function >`  
`sugar::Lapply< RTYPE, NA, T, Function >` [Rcpp::lapply](#) (const [Rcpp::VectorBase](#)< RTYPE, NA, T > &t, Function fun)

## 7.258 inst/include/Rcpp/sugar/functions/Lazy.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

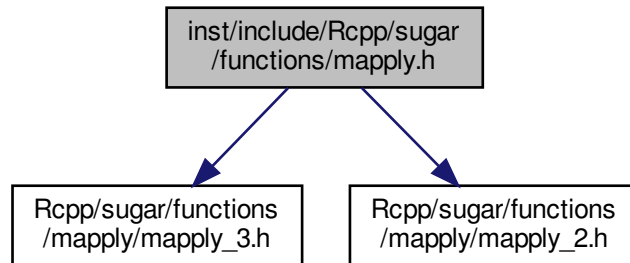
- class [Rcpp::sugar::Lazy< T, EXPR >](#)

### Namespaces

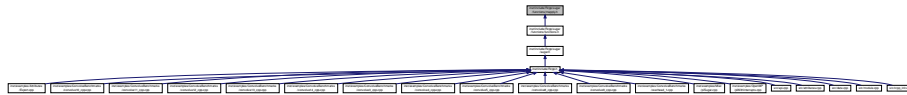
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## 7.259 inst/include/Rcpp/sugar/functions/mapply.h File Reference

```
#include <Rcpp/sugar/functions/mapply/mapply_3.h>
#include <Rcpp/sugar/functions/mapply/mapply_2.h>
Include dependency graph for mapply.h:
```

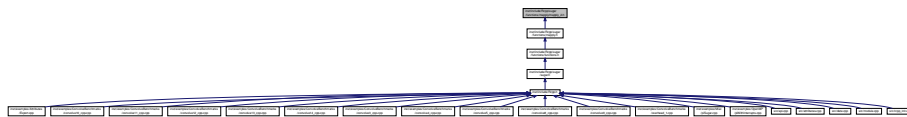


This graph shows which files directly or indirectly include this file:



## 7.260 inst/include/Rcpp/sugar/functions/mapply/mapply\_2.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Mapply\\_2< RTYPE, NA\\_1, T\\_1, NA\\_2, T\\_2, Function >](#)
- class [Rcpp::sugar::Mapply\\_2\\_Vector\\_Primitive< RTYPE, NA\\_1, T\\_1, PRIM\\_2, Function >](#)
- class [Rcpp::sugar::Mapply\\_2\\_Primitive\\_Vector< RTYPE, PRIM\\_1, NA\\_2, T\\_2, Function >](#)

## Namespaces

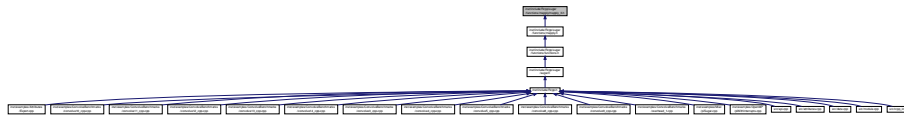
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA_1, typename T_1, bool NA_2, typename T_2, typename Function >`  
`sugar::Mapply_2< RTYPE, NA_1, T_1, NA_2, T_2, Function >` [Rcpp::mapply](#) (const [Rcpp::VectorBase](#)< RTYPE, NA\_1, T\_1 > &t1, const [Rcpp::VectorBase](#)< RTYPE, NA\_2, T\_2 > &t2, Function fun)
- `template<int RTYPE, bool NA_1, typename T_1, typename Function >`  
`sugar::Mapply_2_Vector_Primitive< RTYPE, NA_1, T_1, double, Function >` [Rcpp::mapply](#) (const [Rcpp::VectorBase](#)< RTYPE, NA\_1, T\_1 > &t1, double t2, Function fun)
- `template<int RTYPE, bool NA_2, typename T_2, typename Function >`  
`sugar::Mapply_2_Primitive_Vector< RTYPE, double, NA_2, T_2, Function >` [Rcpp::mapply](#) (double t1, const [Rcpp::VectorBase](#)< RTYPE, NA\_2, T\_2 > &t2, Function fun)

## 7.261 inst/include/Rcpp/sugar/functions/mapply/mapply\_3.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Mapply\\_3](#)< RTYPE\_1, NA\_1, T\_1, RTYPE\_2, NA\_2, T\_2, RTYPE\_3, NA\_3, T\_3, Function >

## Namespaces

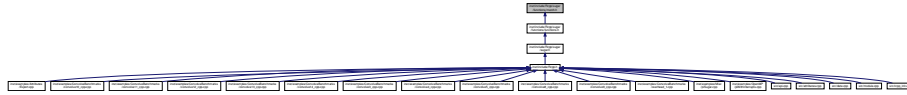
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE_1, bool NA_1, typename T_1, int RTYPE_2, bool NA_2, typename T_2, int RTYPE_3, bool NA_3, typename T_3, typename Function >`  
`sugar::Mapply_3< RTYPE_1, NA_1, T_1, RTYPE_2, NA_2, T_2, RTYPE_3, NA_3, T_3, Function >` [Rcpp::mapply](#) (const [Rcpp::VectorBase](#)< RTYPE\_1, NA\_1, T\_1 > &t1, const [Rcpp::VectorBase](#)< RTYPE\_2, NA\_2, T\_2 > &t2, const [Rcpp::VectorBase](#)< RTYPE\_3, NA\_3, T\_3 > &t3, Function fun)

## 7.262 inst/include/Rcpp/sugar/functions/match.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

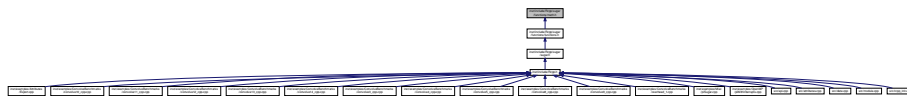
- [Rcpp](#)  
*Rcpp API.*

### Functions

- `template<int RTYPE, bool NA, typename T, bool RHS_NA, typename RHS_T >`  
IntegerVector [Rcpp::match](#) (const VectorBase< RTYPE, NA, T > &x, const VectorBase< RTYPE, RHS\_NA, RHS\_T > &table\_)

## 7.263 inst/include/Rcpp/sugar/functions/math.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

### Functions

- double [Rcpp::internal::factorial](#) (double x)
- double [Rcpp::internal::lfactorial](#) (double x)
- [VECTORIZED\\_MATH\\_1](#) (trunc, ::Rf\_ftrunc) [SUGAR\\_BLOCK\\_2](#)(round

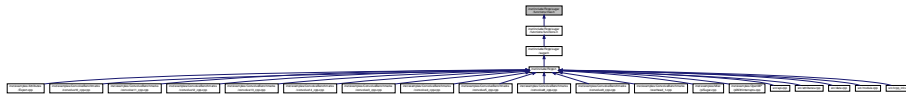
## 7.263.1 Function Documentation

### 7.263.1.1 VECTORIZED\_MATH\_1()

```
VECTORIZED_MATH_1 (
    trunc ,
    ::Rf_ftrunc )
```

## 7.264 inst/include/Rcpp/sugar/functions/max.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Max< RTYPE, NA, T >](#)
- class [Rcpp::sugar::Max< RTYPE, false, T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::Max< RTYPE, NA, T >` [Rcpp::max](#) (const VectorBase< RTYPE, NA, T > &x)

## 7.265 inst/include/Rcpp/sugar/functions/mean.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Mean< RTYPE, NA, T >](#)
- class [Rcpp::sugar::Mean< CPLXSXP, NA, T >](#)
- class [Rcpp::sugar::Mean< LGLSXP, NA, T >](#)
- class [Rcpp::sugar::Mean< INTSXP, NA, T >](#)

## Namespaces

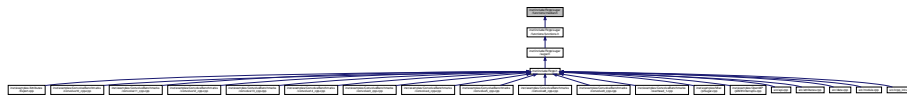
- [Rcpp](#)
  - [Rcpp API.](#)
- [Rcpp::sugar](#)

## Functions

- `template<bool NA, typename T >`  
[sugar::Mean< REALSXP, NA, T >](#) [Rcpp::mean](#) (const VectorBase< REALSXP, NA, T > &t)
- `template<bool NA, typename T >`  
[sugar::Mean< INTSXP, NA, T >](#) [Rcpp::mean](#) (const VectorBase< INTSXP, NA, T > &t)
- `template<bool NA, typename T >`  
[sugar::Mean< CPLXSXP, NA, T >](#) [Rcpp::mean](#) (const VectorBase< CPLXSXP, NA, T > &t)
- `template<bool NA, typename T >`  
[sugar::Mean< LGLSXP, NA, T >](#) [Rcpp::mean](#) (const VectorBase< LGLSXP, NA, T > &t)

## 7.266 inst/include/Rcpp/sugar/functions/median.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::sugar::median\\_detail::result< RTYPE >](#)
- struct [Rcpp::sugar::median\\_detail::result< INTSXP >](#)
- struct [Rcpp::sugar::median\\_detail::result< STRSXP >](#)
- class [Rcpp::sugar::Median< RTYPE, NA, T, NA\\_RM >](#)
- class [Rcpp::sugar::Median< RTYPE, NA, T, true >](#)
- class [Rcpp::sugar::Median< RTYPE, false, T, NA\\_RM >](#)
- class [Rcpp::sugar::Median< STRSXP, NA, T, NA\\_RM >](#)
- class [Rcpp::sugar::Median< STRSXP, NA, T, true >](#)
- class [Rcpp::sugar::Median< STRSXP, false, T, true >](#)

## Namespaces

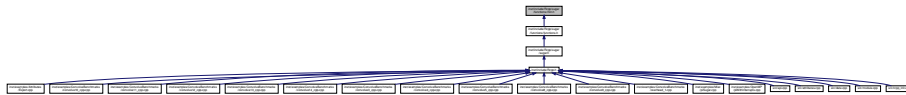
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)
- [Rcpp::sugar::median\\_detail](#)

## Functions

- `template<typename T >`  
`bool Rcpp::sugar::median\_detail::less (T lhs, T rhs)`
- `template<> bool Rcpp::sugar::median\_detail::less< Rcomplex > (Rcomplex lhs, Rcomplex rhs)`
- `double Rcpp::sugar::median\_detail::half (double lhs)`
- `double Rcpp::sugar::median\_detail::half (int lhs)`
- `Rcomplex Rcpp::sugar::median\_detail::half (Rcomplex lhs)`
- `template<int RTYPE, bool NA, typename T >`  
`sugar::median_detail::result< RTYPE >::type Rcpp::median (const Rcpp::VectorBase< RTYPE, NA, T > &x, bool na_rm=false)`

## 7.267 inst/include/Rcpp/sugar/functions/min.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Min](#)< RTYPE, NA, T >
- class [Rcpp::sugar::Min](#)< RTYPE, false, T >

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

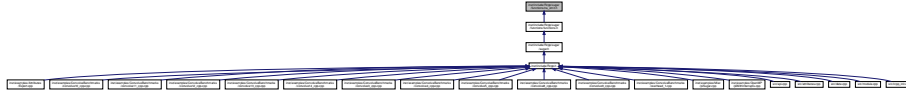
## Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::Min< RTYPE, NA, T > Rcpp::min (const VectorBase< RTYPE, NA, T > &x)`



## 7.268 inst/include/Rcpp/sugar/functions/na\_omit.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

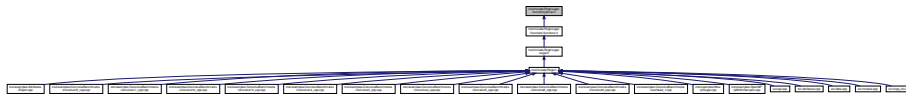
- [Rcpp](#)
  - *Rcpp API*
- [Rcpp::sugar](#)

### Functions

- `template<int RTYPE, bool NA, typename T >`  
`Vector< RTYPE >` [Rcpp::sugar::na\\_omit\\_impl](#) (const T &x, [Rcpp::traits::false\\_type](#))
- `template<int RTYPE, bool NA, typename T >`  
`Vector< RTYPE >` [Rcpp::sugar::na\\_omit\\_impl](#) (const T &x, [Rcpp::traits::true\\_type](#))
- `template<int RTYPE, bool NA, typename T >`  
`Vector< RTYPE >` [Rcpp::na\\_omit](#) (const VectorBase< RTYPE, NA, T > &t)

## 7.269 inst/include/Rcpp/sugar/functions/pmax.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- `struct` [Rcpp::sugar::pmax\\_op< REALSXP, true, true >](#)
- `struct` [Rcpp::sugar::pmax\\_op< REALSXP, true, false >](#)
- `struct` [Rcpp::sugar::pmax\\_op< REALSXP, false, true >](#)
- `struct` [Rcpp::sugar::pmax\\_op< REALSXP, false, false >](#)
- `struct` [Rcpp::sugar::pmax\\_op< INTSXP, LHS\\_NA, RHS\\_NA >](#)
- `class` [Rcpp::sugar::pmax\\_op\\_Vector\\_Primitive< RTYPE, NA >](#)
- `class` [Rcpp::sugar::pmax\\_op\\_Vector\\_Primitive< REALSXP, true >](#)
- `class` [Rcpp::sugar::Pmax\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- `class` [Rcpp::sugar::Pmax\\_Vector\\_Primitive< RTYPE, LHS\\_NA, LHS\\_T >](#)

## Namespaces

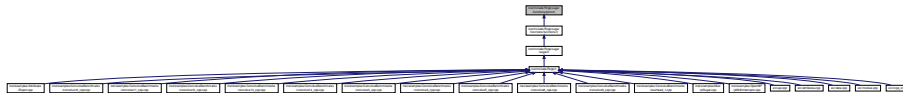
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`sugar::Pmax_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >` [Rcpp::pmax](#) (const [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, const [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T >`  
`sugar::Pmax_Vector_Primitive< RTYPE, LHS_NA, LHS_T >` [Rcpp::pmax](#) (const [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, typename [Rcpp::traits::storage\\_type](#)< RTYPE >::type rhs)
- `template<int RTYPE, bool RHS_NA, typename RHS_T >`  
`sugar::Pmax_Vector_Primitive< RTYPE, RHS_NA, RHS_T >` [Rcpp::pmax](#) (typename [Rcpp::traits::storage\\_type](#)< RTYPE >::type lhs, const [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)

## 7.270 inst/include/Rcpp/sugar/functions/pmin.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::sugar::pmin\\_op](#)< REALSXP, true, true >
- struct [Rcpp::sugar::pmin\\_op](#)< REALSXP, true, false >
- struct [Rcpp::sugar::pmin\\_op](#)< REALSXP, false, true >
- struct [Rcpp::sugar::pmin\\_op](#)< REALSXP, false, false >
- struct [Rcpp::sugar::pmin\\_op](#)< INTSXP, LHS\_NA, RHS\_NA >
- class [Rcpp::sugar::pmin\\_op\\_Vector\\_Primitive](#)< RTYPE, NA >
- class [Rcpp::sugar::pmin\\_op\\_Vector\\_Primitive](#)< REALSXP, true >
- class [Rcpp::sugar::Pmin\\_Vector\\_Vector](#)< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >
- class [Rcpp::sugar::Pmin\\_Vector\\_Primitive](#)< RTYPE, LHS\_NA, LHS\_T >

## Namespaces

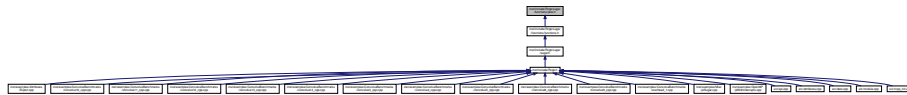
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >`  
`sugar::Pmin_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >` [Rcpp::pmin](#) (const [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, const [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T >`  
`sugar::Pmin_Vector_Primitive< RTYPE, LHS_NA, LHS_T >` [Rcpp::pmin](#) (const [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, typename [Rcpp::traits::storage\\_type](#)< RTYPE >::type rhs)
- `template<int RTYPE, bool RHS_NA, typename RHS_T >`  
`sugar::Pmin_Vector_Primitive< RTYPE, RHS_NA, RHS_T >` [Rcpp::pmin](#) (typename [Rcpp::traits::storage\\_type](#)< RTYPE >::type lhs, const [Rcpp::VectorBase](#)< RTYPE, RHS\_NA, RHS\_T > &rhs)

## 7.271 inst/include/Rcpp/sugar/functions/pow.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Pow](#)< RTYPE, NA, T, EXPONENT\_TYPE >
- class [Rcpp::sugar::Pow](#)< INTSXP, NA, T, EXPONENT\_TYPE >
- class [Rcpp::sugar::Pow](#)< INTSXP, false, T, EXPONENT\_TYPE >

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T , typename EXPONENT_TYPE >`  
`sugar::Pow< RTYPE, NA, T, EXPONENT_TYPE >` [Rcpp::pow](#) (const [VectorBase](#)< RTYPE, NA, T > &t, EXPONENT\_TYPE exponent)

## 7.272 inst/include/Rcpp/sugar/functions/range.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Range< RTYPE, NA, T >](#)
- class [Rcpp::sugar::Range< RTYPE, false, T >](#)

## Namespaces

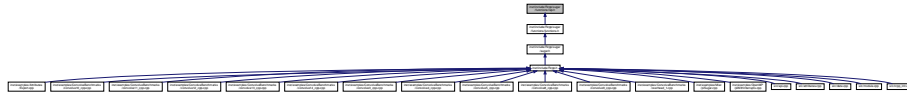
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::Range< RTYPE, NA, T >` [Rcpp::range](#) (const VectorBase< RTYPE, NA, T > &x)

## 7.273 inst/include/Rcpp/sugar/functions/rep.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Rep< RTYPE, NA, T >](#)
- class [Rcpp::sugar::Rep\\_Single< T >](#)

## Namespaces

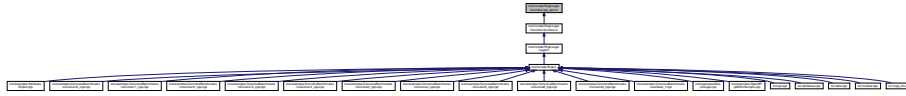
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::Rep< RTYPE, NA, T >` [Rcpp::rep](#) (const VectorBase< RTYPE, NA, T > &t, R\_xlen\_t n)
- `sugar::Rep_Single< double >` [Rcpp::rep](#) (const double &x, R\_xlen\_t n)
- `sugar::Rep_Single< int >` [Rcpp::rep](#) (const int &x, R\_xlen\_t n)
- `sugar::Rep_Single< Rbyte >` [Rcpp::rep](#) (const Rbyte &x, R\_xlen\_t n)
- `sugar::Rep_Single< Rcomplex >` [Rcpp::rep](#) (const Rcomplex &x, R\_xlen\_t n)
- `sugar::Rep_Single< bool >` [Rcpp::rep](#) (const bool &x, R\_xlen\_t n)

## 7.274 inst/include/Rcpp/sugar/functions/rep\_each.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Rep\\_each< RTYPE, NA, T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- `template<int RTYPE, bool NA, typename T >`  
[sugar::Rep\\_each< RTYPE, NA, T >](#) [Rcpp::rep\\_each](#) (const `VectorBase< RTYPE, NA, T >` &t, `R_xlen_t` times)

## 7.275 inst/include/Rcpp/sugar/functions/rep\_len.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Rep\\_len< RTYPE, NA, T >](#)

### Namespaces

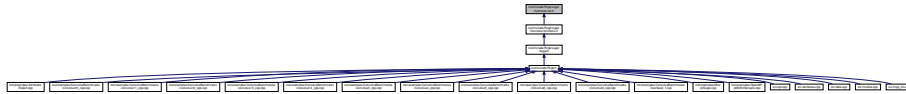
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::Rep_len< RTYPE, NA, T >` [Rcpp::rep\\_len](#) (const VectorBase< RTYPE, NA, T > &t, R\_xlen\_t len)

## 7.276 inst/include/Rcpp/sugar/functions/rev.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Rev< RTYPE, NA, T >](#)

## Namespaces

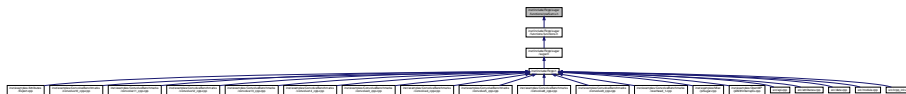
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::Rev< RTYPE, NA, T >` [Rcpp::rev](#) (const VectorBase< RTYPE, NA, T > &t)

## 7.277 inst/include/Rcpp/sugar/functions/rowSums.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::sugar::detail::RowSumsReturn< RTYPE >](#)
- struct [Rcpp::sugar::detail::RowSumsReturn< LGLSXP >](#)
- struct [Rcpp::sugar::detail::ColSumsReturn< RTYPE >](#)
- struct [Rcpp::sugar::detail::RowMeansReturn< RTYPE >](#)
- struct [Rcpp::sugar::detail::RowMeansReturn< CPLXSXP >](#)
- struct [Rcpp::sugar::detail::ColMeansReturn< RTYPE >](#)
- class [Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, NA\\_RM >](#)
- class [Rcpp::sugar::RowSumsImpl< RTYPE, NA, T, true >](#)
- class [Rcpp::sugar::RowSumsImpl< RTYPE, false, T, NA\\_RM >](#)
- class [Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, NA\\_RM >](#)
- class [Rcpp::sugar::ColSumsImpl< RTYPE, NA, T, true >](#)
- class [Rcpp::sugar::ColSumsImpl< RTYPE, false, T, NA\\_RM >](#)
- class [Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, NA\\_RM >](#)
- class [Rcpp::sugar::RowMeansImpl< RTYPE, NA, T, true >](#)
- class [Rcpp::sugar::RowMeansImpl< RTYPE, false, T, NA\\_RM >](#)
- class [Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, NA\\_RM >](#)
- class [Rcpp::sugar::ColMeansImpl< RTYPE, NA, T, true >](#)
- class [Rcpp::sugar::ColMeansImpl< RTYPE, false, T, NA\\_RM >](#)

## Namespaces

- [Rcpp](#)
  - *Rcpp API.*
- [Rcpp::sugar](#)
- [Rcpp::sugar::detail](#)

## Macros

- `#define ROW\_SUMS\_IMPL\_KEEPNA(__RTYPE__)`
- `#define ROW\_SUMS\_IMPL\_RMNA(__RTYPE__)`
- `#define COL\_SUMS\_IMPL\_KEEPNA(__RTYPE__)`
- `#define COL\_SUMS\_IMPL\_RMNA(__RTYPE__)`
- `#define ROW\_MEANS\_IMPL\_KEEPNA(__RTYPE__)`
- `#define ROW\_MEANS\_IMPL\_RMNA(__RTYPE__)`
- `#define COL\_MEANS\_IMPL\_KEEPNA(__RTYPE__)`
- `#define COL\_MEANS\_IMPL\_RMNA(__RTYPE__)`

## Functions

- bool `Rcpp::sugar::detail::check_na` (double x)
- bool `Rcpp::sugar::detail::check_na` (int x)
- bool `Rcpp::sugar::detail::check_na` (Rboolean x)
- bool `Rcpp::sugar::detail::check_na` (SEXP x)
- bool `Rcpp::sugar::detail::check_na` (Rcomplex x)
- void `Rcpp::sugar::detail::incr` (double \*lhs, double rhs)
- void `Rcpp::sugar::detail::incr` (int \*lhs, int rhs)
- void `Rcpp::sugar::detail::incr` (Rcomplex \*lhs, const Rcomplex &rhs)
- void `Rcpp::sugar::detail::div` (double \*lhs, R\_xlen\_t rhs)
- void `Rcpp::sugar::detail::div` (Rcomplex \*lhs, R\_xlen\_t rhs)
- void `Rcpp::sugar::detail::set_nan` (double \*x)
- void `Rcpp::sugar::detail::set_nan` (Rcomplex \*x)
- template<int RTYPE, bool NA, typename T >  
`sugar::detail::RowSumsReturn< RTYPE >::type Rcpp::rowSums` (const MatrixBase< RTYPE, NA, T > &x, bool na\_rm=false)
- template<int RTYPE, bool NA, typename T >  
`sugar::detail::ColSumsReturn< RTYPE >::type Rcpp::colSums` (const MatrixBase< RTYPE, NA, T > &x, bool na\_rm=false)
- template<int RTYPE, bool NA, typename T >  
`sugar::detail::RowMeansReturn< RTYPE >::type Rcpp::rowMeans` (const MatrixBase< RTYPE, NA, T > &x, bool na\_rm=false)
- template<int RTYPE, bool NA, typename T >  
`sugar::detail::ColMeansReturn< RTYPE >::type Rcpp::colMeans` (const MatrixBase< RTYPE, NA, T > &x, bool na\_rm=false)

### 7.277.1 Macro Definition Documentation

#### 7.277.1.1 COL\_MEANS\_IMPL\_KEEPNA

```
#define COL_MEANS_IMPL_KEEPNA(  
    __RTYPE__ )
```

Definition at line 753 of file rowSums.h.

#### 7.277.1.2 COL\_MEANS\_IMPL\_RMNA

```
#define COL_MEANS_IMPL_RMNA(  
    __RTYPE__ )
```

Definition at line 861 of file rowSums.h.



### 7.277.1.3 COL\_SUMS\_IMPL\_KEEPNA

```
#define COL_SUMS_IMPL_KEEPNA(  
    __RTYPE__ )
```

Definition at line 353 of file rowSums.h.

### 7.277.1.4 COL\_SUMS\_IMPL\_RMNA

```
#define COL_SUMS_IMPL_RMNA(  
    __RTYPE__ )
```

Definition at line 448 of file rowSums.h.

### 7.277.1.5 ROW\_MEANS\_IMPL\_KEEPNA

```
#define ROW_MEANS_IMPL_KEEPNA(  
    __RTYPE__ )
```

Definition at line 544 of file rowSums.h.

### 7.277.1.6 ROW\_MEANS\_IMPL\_RMNA

```
#define ROW_MEANS_IMPL_RMNA(  
    __RTYPE__ )
```

Definition at line 652 of file rowSums.h.

### 7.277.1.7 ROW\_SUMS\_IMPL\_KEEPNA

```
#define ROW_SUMS_IMPL_KEEPNA(  
    __RTYPE__ )
```

Definition at line 169 of file rowSums.h.

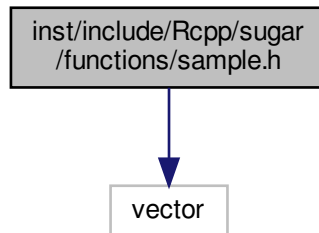
### 7.277.1.8 ROW\_SUMS\_IMPL\_RMNA

```
#define ROW_SUMS_IMPL_RMNA(  
    __RTYPE__ )
```

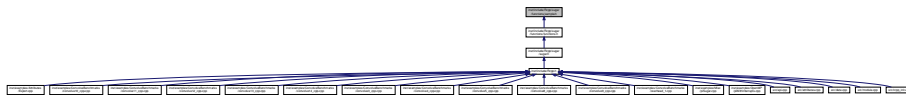
Definition at line 264 of file rowSums.h.

## 7.278 inst/include/Rcpp/sugar/functions/sample.h File Reference

```
#include <vector>  
Include dependency graph for sample.h:
```



This graph shows which files directly or indirectly include this file:



### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Typedefs

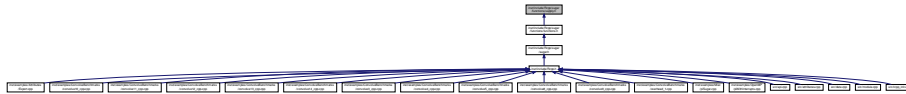
- typedef Nullable< Vector< REALSXP > > [Rcpp::sugar::probs\\_t](#)

## Functions

- void [Rcpp::sugar::Normalize](#) (Vector< REALSXP > &p, int require\_k, bool replace)
- Vector< INTSXP > [Rcpp::sugar::SampleReplace](#) (Vector< REALSXP > &p, int n, int k, bool one\_based)
- template<int RTYPE>  
Vector< RTYPE > [Rcpp::sugar::SampleReplace](#) (Vector< REALSXP > &p, int k, const Vector< RTYPE > &ref)
- Vector< INTSXP > [Rcpp::sugar::WalkerSample](#) (const Vector< REALSXP > &p, int n, int nans, bool one\_based)
- template<int RTYPE>  
Vector< RTYPE > [Rcpp::sugar::WalkerSample](#) (const Vector< REALSXP > &p, int nans, const Vector< RTYPE > &ref)
- Vector< INTSXP > [Rcpp::sugar::SampleNoReplace](#) (Vector< REALSXP > &p, int n, int nans, bool one\_based)
- template<int RTYPE>  
Vector< RTYPE > [Rcpp::sugar::SampleNoReplace](#) (Vector< REALSXP > &p, int nans, const Vector< RTYPE > &ref)
- Vector< INTSXP > [Rcpp::sugar::EmpiricalSample](#) (int n, int size, bool replace, bool one\_based)
- template<int RTYPE>  
Vector< RTYPE > [Rcpp::sugar::EmpiricalSample](#) (int size, bool replace, const Vector< RTYPE > &ref)
- Vector< INTSXP > [Rcpp::sample](#) (int n, int size, bool replace=false, sugar::probs\_t probs=R\_NilValue, bool one\_based=true)
- template<int RTYPE>  
Vector< RTYPE > [Rcpp::sample](#) (const Vector< RTYPE > &x, int size, bool replace=false, sugar::probs\_t probs=R\_NilValue)

## 7.279 inst/include/Rcpp/sugar/functions/sapply.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::sugar::sapply\\_application\\_result\\_of< Function, SugarExpression >](#)
- class [Rcpp::sugar::Sapply< RTYPE, NA, T, Function, NO\\_CONVERSION >](#)
- class [Rcpp::sugar::Sapply< RTYPE, NA, T, Function, true >](#)

## Namespaces

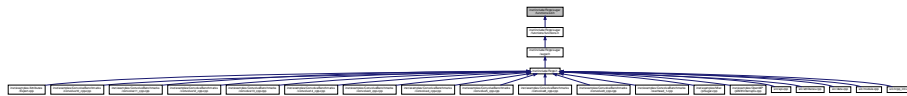
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T, typename Function >`  
`sugar::Sapply< RTYPE, NA, T, Function, traits::same_type< typename ::Rcpp::sugar::sapply_application_result_of<`  
`Function, T >::type, typename Rcpp::traits::storage_type< traits::r_sexptype_traits< typename ::Rcpp::sugar::sapply_application_r`  
`Function, T >::type >::rtype >::type >::value > Rcpp::sapply (const Rcpp::VectorBase< RTYPE, NA, T > &t,`  
`Function fun)`

## 7.280 inst/include/Rcpp/sugar/functions/sd.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class `Rcpp::sugar::Sd< RTYPE, NA, T >`

## Namespaces

- `Rcpp`  
*Rcpp API.*
- `Rcpp::sugar`

## Functions

- `template<bool NA, typename T >`  
`sugar::Sd< REALSXP, NA, T > Rcpp::sd (const VectorBase< REALSXP, NA, T > &t)`

## 7.281 inst/include/Rcpp/sugar/functions/self\_match.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::SelfInserter< HASH, STORAGE >](#)
- class [Rcpp::sugar::SelfMatch< RTYPE, TABLE\\_T >](#)

## Namespaces

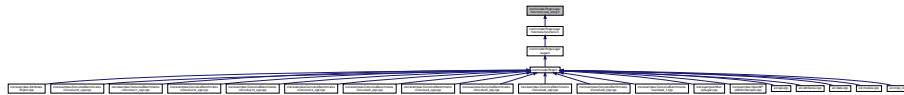
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- [template<int RTYPE, bool NA, typename T >](#)  
[IntegerVector Rcpp::self\\_match](#) (const [VectorBase< RTYPE, NA, T >](#) &x)

## 7.282 inst/include/Rcpp/sugar/functions/seq\_along.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::SeqLen](#)

## Namespaces

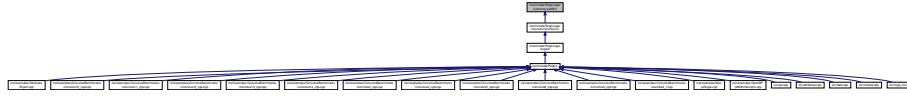
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- [template<int RTYPE, bool NA, typename T >](#)  
[sugar::SeqLen Rcpp::seq\\_along](#) (const [Rcpp::VectorBase< RTYPE, NA, T >](#) &t)
- [sugar::SeqLen Rcpp::seq\\_len](#) (const [size\\_t](#) &n)
- [Range Rcpp::seq](#) ([R\\_xlen\\_t](#) start, [R\\_xlen\\_t](#) end)

## 7.283 inst/include/Rcpp/sugar/functions/setdiff.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::RemoveFromSet< SET >](#)
- class [Rcpp::sugar::SetDiff< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::SetEqual< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Intersect< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Union< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`Vector< RTYPE > Rcpp::setdiff (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)`
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`bool Rcpp::setequal (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)`
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`Vector< RTYPE > Rcpp::intersect (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)`
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`Vector< RTYPE > Rcpp::union\_ (const VectorBase< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)`

## 7.284 inst/include/Rcpp/sugar/functions/sign.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::sign\\_\\_impl< NA, RTYPE >](#)
- class [Rcpp::sugar::sign\\_\\_impl< false, RTYPE >](#)
- class [Rcpp::sugar::Sign< RTYPE, NA, T >](#)

## Namespaces

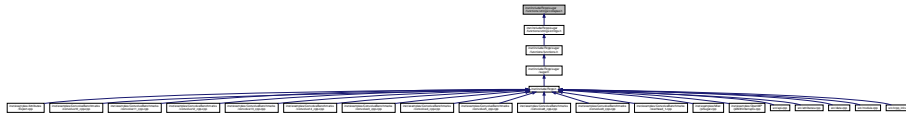
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- [template<bool NA, typename T >](#)  
[sugar::Sign< INTSXP, NA, T >](#) [Rcpp::sign](#) (const [VectorBase< INTSXP, NA, T >](#) &t)
- [template<bool NA, typename T >](#)  
[sugar::Sign< REALSXP, NA, T >](#) [Rcpp::sign](#) (const [VectorBase< REALSXP, NA, T >](#) &t)

## 7.285 inst/include/Rcpp/sugar/functions/strings/collapse.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

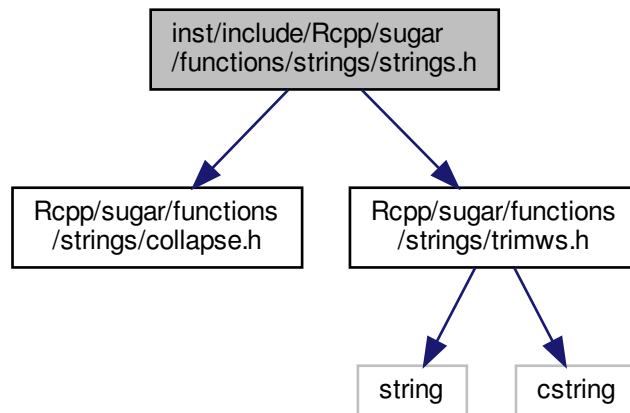
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

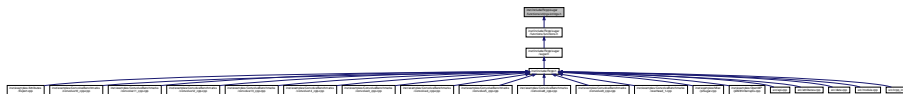
- [template<typename Iterator >](#)  
String [Rcpp::sugar::collapse\\_\\_impl](#) (Iterator it, R\_xlen\_t n)
- [template<bool NA, typename T >](#)  
String [Rcpp::collapse](#) (const [VectorBase< STRSXP, NA, T >](#) &vec)

## 7.286 inst/include/Rcpp/sugar/functions/strings/strings.h File Reference

```
#include <Rcpp/sugar/functions/strings/collapse.h>
#include <Rcpp/sugar/functions/strings/trimws.h>
Include dependency graph for strings.h:
```



This graph shows which files directly or indirectly include this file:

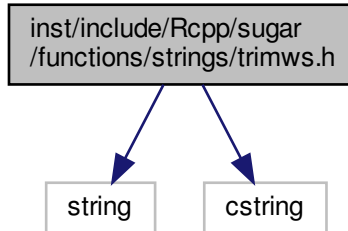


## 7.287 inst/include/Rcpp/sugar/functions/strings/trimws.h File Reference

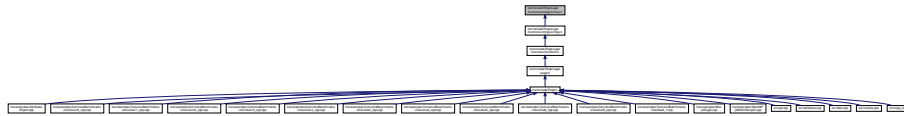
```
#include <string>
#include <cstring>
```



Include dependency graph for trimws.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)
- [Rcpp::sugar::detail](#)

## Functions

- `bool Rcpp::sugar::detail::isws (const char c)`
- `const char * Rcpp::sugar::detail::trim\_left (const char *str)`
- `const char * Rcpp::sugar::detail::trim\_right (const char *str, R_len_t sz, std::string *buff)`
- `const char * Rcpp::sugar::detail::trim\_both (const char *str, R_len_t sz, std::string *buff)`
- `Vector< STRSXP > Rcpp::trimws (const Vector< STRSXP > &x, const char *which="both")`
- `Matrix< STRSXP > Rcpp::trimws (const Matrix< STRSXP > &x, const char *which="both")`
- `String Rcpp::trimws (const String &str, const char *which="both")`

## 7.288 inst/include/Rcpp/sugar/functions/sum.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Sum< RTYPE, NA, T >](#)
- class [Rcpp::sugar::Sum< REALSXP, NA, T >](#)
- class [Rcpp::sugar::Sum< RTYPE, false, T >](#)

## Namespaces

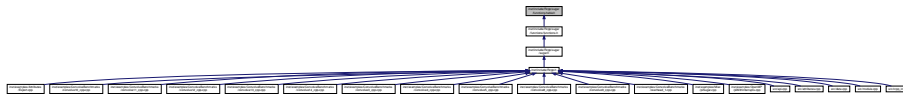
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<bool NA, typename T >`  
`sugar::Sum< INTSXP, NA, T >` [Rcpp::sum](#) (const VectorBase< INTSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Sum< REALSXP, NA, T >` [Rcpp::sum](#) (const VectorBase< REALSXP, NA, T > &t)
- `template<bool NA, typename T >`  
`sugar::Sum< LGLSXP, NA, T >` [Rcpp::sum](#) (const VectorBase< LGLSXP, NA, T > &t)

## 7.289 inst/include/Rcpp/sugar/functions/table.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::CountInserter< HASH, STORAGE >](#)
- class [Rcpp::sugar::Grabber< HASH, RTYPE >](#)
- class [Rcpp::sugar::Table< RTYPE, TABLE\\_T >](#)

## Namespaces

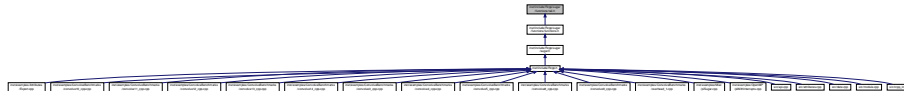
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T >`  
IntegerVector [Rcpp::table](#) (const VectorBase< RTYPE, NA, T > &x)

## 7.290 inst/include/Rcpp/sugar/functions/tail.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Tail](#)< RTYPE, NA, T >

## Namespaces

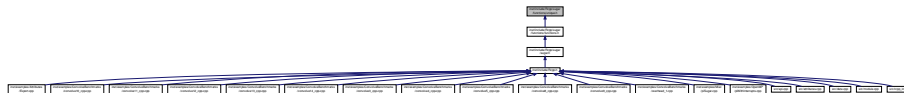
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::Tail`< RTYPE, NA, T > [Rcpp::tail](#) (const VectorBase< RTYPE, NA, T > &t, R\_xlen\_t n)

## 7.291 inst/include/Rcpp/sugar/functions/unique.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::InSet](#)< HASH >
- class [Rcpp::sugar::In](#)< RTYPE, TABLE\_T >

## Namespaces

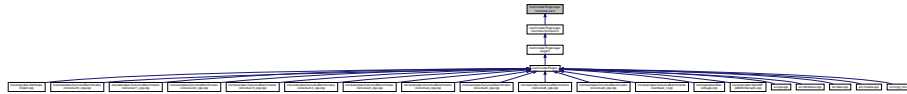
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T >`  
`Vector< RTYPE >` [Rcpp::unique](#) (`const VectorBase< RTYPE, NA, T > &t`)
- `template<int RTYPE, bool NA, typename T >`  
`Vector< RTYPE >` [Rcpp::sort\\_unique](#) (`const VectorBase< RTYPE, NA, T > &t`, `bool decreasing=false`)
- `template<int RTYPE, bool NA, typename T, bool RHS_NA, typename RHS_T >`  
`LogicalVector` [Rcpp::in](#) (`const VectorBase< RTYPE, NA, T > &x`, `const VectorBase< RTYPE, RHS_NA, RHS_T > &table`)

## 7.292 inst/include/Rcpp/sugar/functions/var.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Var< RTYPE, NA, T >](#)
- class [Rcpp::sugar::Var< CPLXSXP, NA, T >](#)

## Namespaces

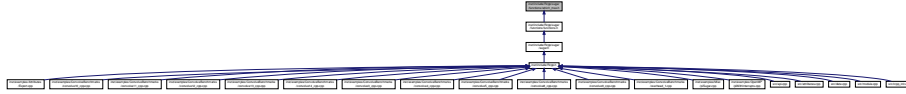
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<bool NA, typename T >`  
`sugar::Var< REALSXP, NA, T >` [Rcpp::var](#) (`const VectorBase< REALSXP, NA, T > &t`)
- `template<bool NA, typename T >`  
`sugar::Var< INTSXP, NA, T >` [Rcpp::var](#) (`const VectorBase< INTSXP, NA, T > &t`)
- `template<bool NA, typename T >`  
`sugar::Var< LGLSXP, NA, T >` [Rcpp::var](#) (`const VectorBase< LGLSXP, NA, T > &t`)
- `template<bool NA, typename T >`  
`sugar::Var< CPLXSXP, NA, T >` [Rcpp::var](#) (`const VectorBase< CPLXSXP, NA, T > &t`)

## 7.293 inst/include/Rcpp/sugar/functions/which\_max.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::WhichMax< RTYPE, NA, T >](#)
- class [Rcpp::sugar::WhichMax< RTYPE, false, T >](#)

### Namespaces

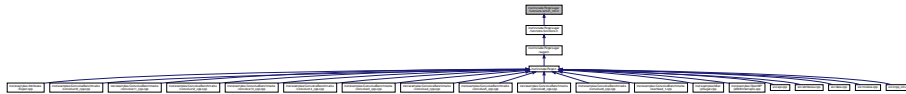
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- [template<int RTYPE, bool NA, typename T >](#)  
[R\\_xlen\\_t Rcpp::which\\_max](#) (const [VectorBase< RTYPE, NA, T >](#) &t)

## 7.294 inst/include/Rcpp/sugar/functions/which\_min.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::WhichMin< RTYPE, NA, T >](#)
- class [Rcpp::sugar::WhichMin< RTYPE, false, T >](#)

### Namespaces

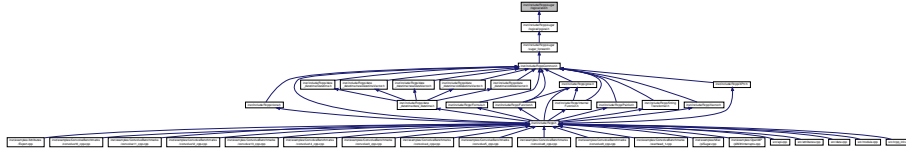
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T >`  
`R_xlen_t Rcpp::which_min (const VectorBase< RTYPE, NA, T > &t)`

## 7.295 inst/include/Rcpp/sugar/logical/and.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class `Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T >`
- class `Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, false, RHS_T >`
- class `Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, RHS_NA, RHS_T >`
- class `Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< false, LHS_T, false, RHS_T >`
- class `Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T >`
- class `Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T >`
- class `Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, RHS_NA, RHS_T >`
- class `Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, false, RHS_T >`
- class `Rcpp::sugar::And_LogicalExpression_LogicalExpression< false, LHS_T, false, RHS_T >`

## Namespaces

- `Rcpp`  
*Rcpp API.*
- `Rcpp::sugar`

## Functions

- `template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult< LHS_NA, LHS_T, RHS_NA, RHS_T > operator&&`  
`(const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > &lhs, const Rcpp::sugar::SingleLogicalResult<`  
`LHS_NA, LHS_T > &rhs)`
- `template<bool LHS_NA, typename LHS_T >`  
`Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T > operator&& (const Rcpp::sugar::SingleLogicalResult<`  
`LHS_NA, LHS_T > &lhs, bool rhs)`
- `template<bool LHS_NA, typename LHS_T >`  
`Rcpp::sugar::And_SingleLogicalResult_bool< LHS_NA, LHS_T > operator&& (bool rhs, const Rcpp::sugar::SingleLogicalResult<`  
`LHS_NA, LHS_T > &lhs)`
- `template<bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`Rcpp::sugar::And_LogicalExpression_LogicalExpression< LHS_NA, LHS_T, RHS_NA, RHS_T > operator&`  
`(const Rcpp::VectorBase< LGLSXP, LHS_NA, LHS_T > &lhs, const Rcpp::VectorBase< LGLSXP, RHS_NA,`  
`RHS_T > &rhs)`

## 7.295.1 Function Documentation

### 7.295.1.1 operator&()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::And_LogicalExpression_LogicalExpression<LHS_NA,LHS_T,RHS_NA,RHS_T> operator& (
    const Rcpp::VectorBase< LGLSXP, LHS_NA, LHS_T > & lhs,
    const Rcpp::VectorBase< LGLSXP, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 316 of file and.h.

### 7.295.1.2 operator&&() [1/3]

```
template<bool LHS_NA, typename LHS_T >
Rcpp::sugar::And_SingleLogicalResult_bool<LHS_NA,LHS_T> operator&& (
    bool rhs,
    const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & lhs ) [inline]
```

Definition at line 306 of file and.h.

### 7.295.1.3 operator&&() [2/3]

```
template<bool LHS_NA, typename LHS_T >
Rcpp::sugar::And_SingleLogicalResult_bool<LHS_NA,LHS_T> operator&& (
    const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & lhs,
    bool rhs ) [inline]
```

Definition at line 297 of file and.h.

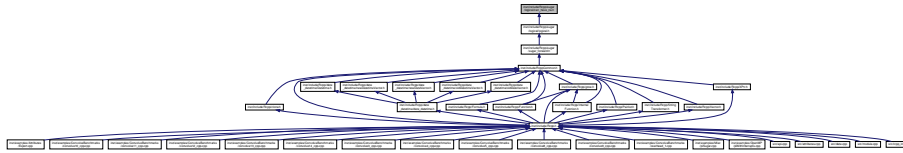
### 7.295.1.4 operator&&() [3/3]

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::And_SingleLogicalResult_SingleLogicalResult<LHS_NA,LHS_T,RHS_NA,RHS_T> operator&& (
    const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & lhs,
    const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & rhs ) [inline]
```

Definition at line 288 of file and.h.

## 7.296 inst/include/Rcpp/sugar/logical/can\_have\_na.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::can\\_have\\_na< T >](#)

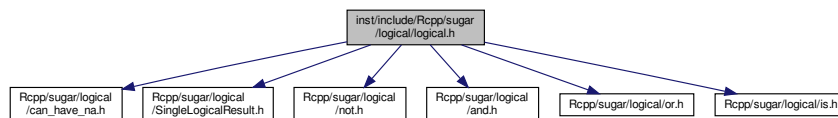
### Namespaces

- [Rcpp](#)  
*Rcpp API.*

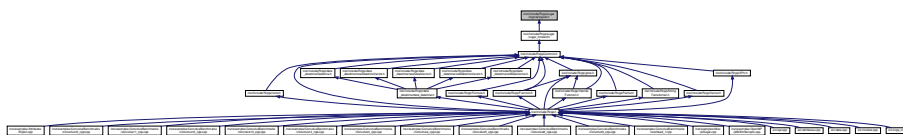
## 7.297 inst/include/Rcpp/sugar/logical/logical.h File Reference

```
#include <Rcpp/sugar/logical/can_have_na.h>
#include <Rcpp/sugar/logical/SingleLogicalResult.h>
#include <Rcpp/sugar/logical/not.h>
#include <Rcpp/sugar/logical/and.h>
#include <Rcpp/sugar/logical/or.h>
#include <Rcpp/sugar/logical/is.h>
```

Include dependency graph for logical.h:



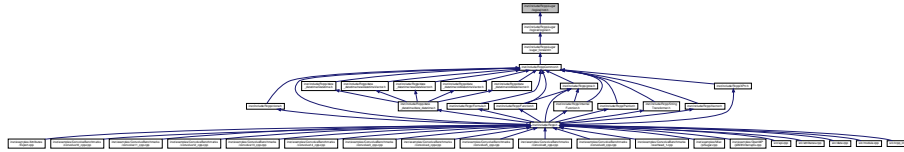
This graph shows which files directly or indirectly include this file:





## 7.298 inst/include/Rcpp/sugar/logical/not.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::sugar::negate< NA >](#)
- struct [Rcpp::sugar::negate< false >](#)
- class [Rcpp::sugar::Negate\\_SingleLogicalResult< NA, T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- `template<bool NA, typename T >`  
`Rcpp::sugar::Negate\_SingleLogicalResult< NA, T > operator! (const Rcpp::sugar::SingleLogicalResult< NA, T > &x)`

## 7.298.1 Function Documentation

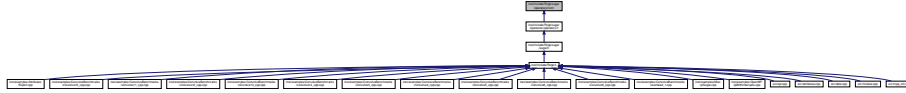
### 7.298.1.1 operator"!()

```
template<bool NA, typename T >
Rcpp::sugar::Negate_SingleLogicalResult<NA,T> operator! (
    const Rcpp::sugar::SingleLogicalResult< NA, T > & x ) [inline]
```

Definition at line 62 of file not.h.

## 7.299 inst/include/Rcpp/sugar/operators/not.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::not\\_< RTYPE, NA >](#)
- class [Rcpp::sugar::not\\_< RTYPE, false >](#)
- class [Rcpp::sugar::not\\_< REALSXP, NA >](#)
- class [Rcpp::sugar::not\\_< REALSXP, false >](#)
- class [Rcpp::sugar::not\\_< CPLXSXP, NA >](#)
- class [Rcpp::sugar::not\\_< CPLXSXP, false >](#)
- class [Rcpp::sugar::Not\\_Vector< RTYPE, NA, T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- [template<int RTYPE, bool NA, typename T > Rcpp::sugar::Not\\_Vector< RTYPE, NA, T > operator! \(const Rcpp::VectorBase< RTYPE, NA, T > &x\)](#)

### 7.299.1 Function Documentation

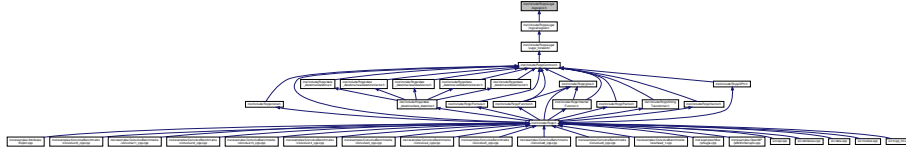
#### 7.299.1.1 operator"!()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Not_Vector< RTYPE , NA , T > operator! (
    const Rcpp::VectorBase< RTYPE, NA, T > & x ) [inline]
```

Definition at line 101 of file not.h.

## 7.300 inst/include/Rcpp/sugar/logical/or.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Or\\_SingleLogicalResult\\_SingleLogicalResult< LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Or\\_SingleLogicalResult\\_SingleLogicalResult< LHS\\_NA, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Or\\_SingleLogicalResult\\_SingleLogicalResult< false, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Or\\_SingleLogicalResult\\_SingleLogicalResult< false, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Or\\_SingleLogicalResult\\_bool< LHS\\_NA, LHS\\_T >](#)
- class [Rcpp::sugar::Or\\_LogicalExpression\\_LogicalExpression< LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Or\\_LogicalExpression\\_LogicalExpression< false, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Or\\_LogicalExpression\\_LogicalExpression< LHS\\_NA, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Or\\_LogicalExpression\\_LogicalExpression< false, LHS\\_T, false, RHS\\_T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- [template<bool LHS\\_NA, typename LHS\\_T, bool RHS\\_NA, typename RHS\\_T > Rcpp::sugar::Or\\_SingleLogicalResult\\_SingleLogicalResult< LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T > operator| \(const Rcpp::sugar::SingleLogicalResult< LHS\\_NA, LHS\\_T > &lhs, const Rcpp::sugar::SingleLogicalResult< LHS\\_NA, LHS\\_T > &rhs\)](#)
- [template<bool LHS\\_NA, typename LHS\\_T > Rcpp::sugar::Or\\_SingleLogicalResult\\_bool< LHS\\_NA, LHS\\_T > operator| \(const Rcpp::sugar::SingleLogicalResult< LHS\\_NA, LHS\\_T > &lhs, bool rhs\)](#)
- [template<bool LHS\\_NA, typename LHS\\_T > Rcpp::sugar::Or\\_SingleLogicalResult\\_bool< LHS\\_NA, LHS\\_T > operator| \(bool rhs, const Rcpp::sugar::SingleLogicalResult< LHS\\_NA, LHS\\_T > &lhs\)](#)
- [template<bool LHS\\_NA, typename LHS\\_T, bool RHS\\_NA, typename RHS\\_T > Rcpp::sugar::Or\\_LogicalExpression\\_LogicalExpression< LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T > operator| \(const Rcpp::VectorBase< LGLSXP, LHS\\_NA, LHS\\_T > &lhs, const Rcpp::VectorBase< LGLSXP, RHS\\_NA, RHS\\_T > &rhs\)](#)

## 7.300.1 Function Documentation

### 7.300.1.1 operator" | ()

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Or_LogicalExpression_LogicalExpression<LHS_NA,LHS_T,RHS_NA,RHS_T> operator| (
    const Rcpp::VectorBase< LGLSXP, LHS_NA, LHS_T > & lhs,
    const Rcpp::VectorBase< LGLSXP, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 315 of file or.h.

### 7.300.1.2 operator" | " | () [1/3]

```
template<bool LHS_NA, typename LHS_T >
Rcpp::sugar::Or_SingleLogicalResult_bool<LHS_NA,LHS_T> operator|| (
    bool rhs,
    const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & lhs ) [inline]
```

Definition at line 305 of file or.h.

### 7.300.1.3 operator" | " | () [2/3]

```
template<bool LHS_NA, typename LHS_T >
Rcpp::sugar::Or_SingleLogicalResult_bool<LHS_NA,LHS_T> operator|| (
    const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & lhs,
    bool rhs ) [inline]
```

Definition at line 296 of file or.h.

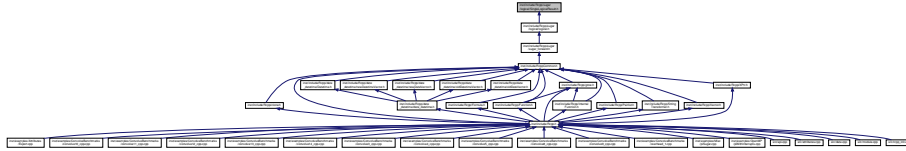
### 7.300.1.4 operator" | " | () [3/3]

```
template<bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Or_SingleLogicalResult_SingleLogicalResult<LHS_NA,LHS_T,RHS_NA,RHS_T> operator|| (
    const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & lhs,
    const Rcpp::sugar::SingleLogicalResult< LHS_NA, LHS_T > & rhs ) [inline]
```

Definition at line 287 of file or.h.

## 7.301 inst/include/Rcpp/sugar/logical/SingleLogicalResult.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

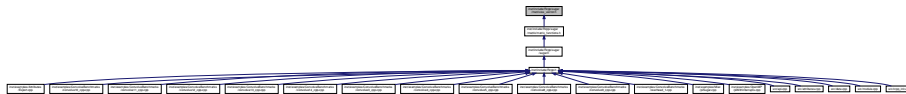
- class [Rcpp::sugar::forbidden\\_conversion< true >](#)
- class [Rcpp::sugar::conversion\\_to\\_bool\\_is\\_forbidden< x >](#)
- class [Rcpp::sugar::SingleLogicalResult< NA, T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## 7.302 inst/include/Rcpp/sugar/matrix/as\_vector.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

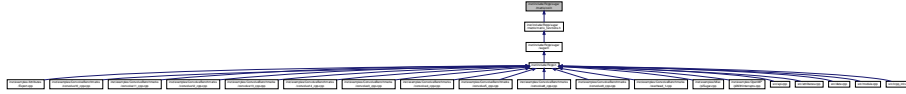
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

### Functions

- [template<int RTYPE, bool NA, typename T >](#)  
[Rcpp::Vector< RTYPE >](#) [Rcpp::internal::as\\_vector\\_impl](#) ([MatrixBase< RTYPE, NA, T >](#) &t, [Rcpp::traits::false\\_type](#))
- [template<int RTYPE, bool NA, typename T >](#)  
[Rcpp::Vector< RTYPE >](#) [Rcpp::internal::as\\_vector\\_impl](#) ([MatrixBase< RTYPE, NA, T >](#) &t, [Rcpp::traits::true\\_type](#))
- [template<int RTYPE, bool NA, typename T >](#)  
[Rcpp::Vector< RTYPE >](#) [Rcpp::as\\_vector](#) ([const MatrixBase< RTYPE, NA, T >](#) &t)

## 7.303 inst/include/Rcpp/sugar/matrix/col.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Col< RTYPE, LHS\\_NA, LHS\\_T >](#)

### Namespaces

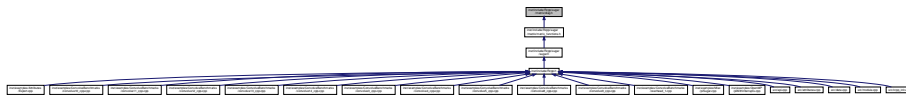
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- [template<int RTYPE, bool LHS\\_NA, typename LHS\\_T > sugar::Col< RTYPE, LHS\\_NA, LHS\\_T > \[Rcpp::col\]\(#\) \(const \[Rcpp::MatrixBase< RTYPE, LHS\\\_NA, LHS\\\_T >\]\(#\) &lhs\)](#)

## 7.304 inst/include/Rcpp/sugar/matrix/diag.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Diag\\_Extractor< RTYPE, NA, T >](#)
- class [Rcpp::sugar::Diag\\_Maker< RTYPE, NA, T >](#)
- struct [Rcpp::sugar::diag\\_result\\_type\\_trait< T >](#)

### Namespaces

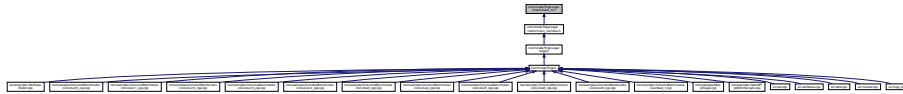
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<typename T >`  
`sugar::diag_result_type_trait< T >::type` [Rcpp::diag](#) (const T &t)

## 7.305 inst/include/Rcpp/sugar/matrix/lower\_tri.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::LowerTri](#)< RTYPE, NA, T >

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

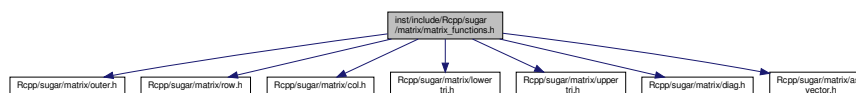
## Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::LowerTri`< RTYPE, NA, T > [Rcpp::lower\\_tri](#) (const [Rcpp::MatrixBase](#)< RTYPE, NA, T > &lhs, bool diag=false)

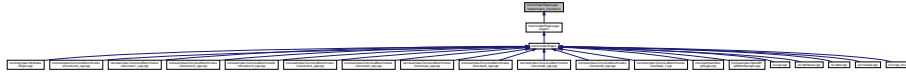
## 7.306 inst/include/Rcpp/sugar/matrix/matrix\_functions.h File Reference

```
#include <Rcpp/sugar/matrix/outer.h>
#include <Rcpp/sugar/matrix/row.h>
#include <Rcpp/sugar/matrix/col.h>
#include <Rcpp/sugar/matrix/lower_tri.h>
#include <Rcpp/sugar/matrix/upper_tri.h>
#include <Rcpp/sugar/matrix/diag.h>
#include <Rcpp/sugar/matrix/as_vector.h>
```

Include dependency graph for `matrix_functions.h`:

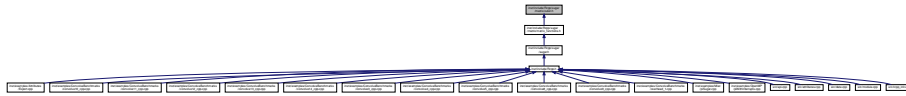


This graph shows which files directly or indirectly include this file:



## 7.307 inst/include/Rcpp/sugar/matrix/outer.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Outer< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T, Function >](#)

### Namespaces

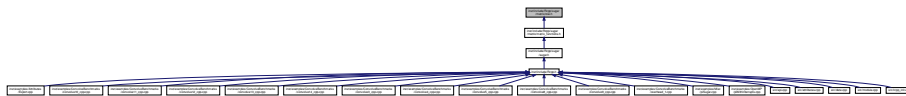
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- [template<int RTYPE, bool LHS\\_NA, typename LHS\\_T, bool RHS\\_NA, typename RHS\\_T, typename Function > sugar::Outer< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T, Function > Rcpp::outer \(const \[Rcpp::VectorBase< RTYPE, LHS\\\_NA, LHS\\\_T >\]\(#\) &lhs, const \[Rcpp::VectorBase< RTYPE, RHS\\\_NA, RHS\\\_T >\]\(#\) &rhs, Function fun\)](#)

## 7.308 inst/include/Rcpp/sugar/matrix/row.h File Reference

This graph shows which files directly or indirectly include this file:





## Classes

- class [Rcpp::sugar::Row< RTYPE, LHS\\_NA, LHS\\_T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool LHS_NA, typename LHS_T >`  
`sugar::Row< RTYPE, LHS_NA, LHS_T >` [Rcpp::row](#) (const [Rcpp::MatrixBase< RTYPE, LHS\\_NA, LHS\\_T >](#) &lhs)

## 7.309 inst/include/Rcpp/sugar/matrix/tools.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

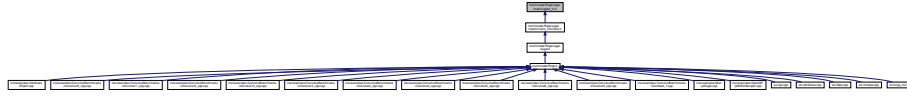
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Functions

- `int` [Rcpp::internal::get\\_line](#) (int index, int nr)
- `int` [Rcpp::internal::get\\_column](#) (int index, int nr)
- `int` [Rcpp::internal::get\\_column](#) (int index, int nr, int i)

## 7.310 inst/include/Rcpp/sugar/matrix/upper\_tri.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::UpperTri< RTYPE, NA, T >](#)

### Namespaces

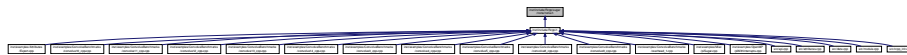
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- `template<int RTYPE, bool NA, typename T >`  
`sugar::UpperTri< RTYPE, NA, T >` [Rcpp::upper\\_tri](#) (const [Rcpp::MatrixBase< RTYPE, NA, T >](#) &lhs, bool diag=false)

## 7.311 inst/include/Rcpp/sugar/nona/nona.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Nona< RTYPE, NA, VECTOR >](#)
- class [Rcpp::sugar::Nona< RTYPE, NA, Rcpp::Vector< RTYPE > >](#)
- class [Rcpp::sugar::NonaPrimitive< T >](#)

### Namespaces

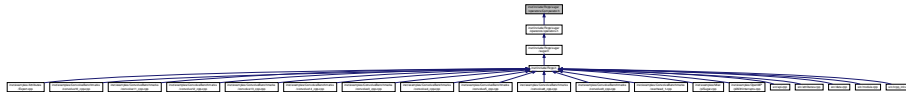
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename VECTOR >`  
`sugar::Nona< RTYPE, NA, VECTOR >` [Rcpp::noNA](#) (`const Rcpp::VectorBase< RTYPE, NA, VECTOR > &vec`)
- `sugar::NonaPrimitive< double >` [Rcpp::noNA](#) (`double x`)
- `sugar::NonaPrimitive< int >` [Rcpp::noNA](#) (`int x`)

## 7.312 inst/include/Rcpp/sugar/operators/Comparator.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Comparator](#)< RTYPE, Operator, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >
- class [Rcpp::sugar::Comparator](#)< RTYPE, Operator, false, LHS\_T, RHS\_NA, RHS\_T >
- class [Rcpp::sugar::Comparator](#)< RTYPE, Operator, false, LHS\_T, false, RHS\_T >

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## 7.313 inst/include/Rcpp/sugar/operators/Comparator\_With\_One\_Value.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

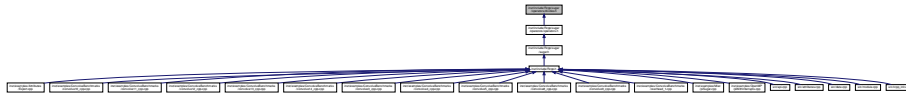
- class [Rcpp::sugar::Comparator\\_With\\_One\\_Value](#)< RTYPE, Operator, NA, T >
- class [Rcpp::sugar::Comparator\\_With\\_One\\_Value](#)< RTYPE, Operator, false, T >

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## 7.314 inst/include/Rcpp/sugar/operators/divides.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Divides\\_Vector\\_Vector](#)< RTYPE, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >
- class [Rcpp::sugar::Divides\\_Vector\\_Vector](#)< REALSXP, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >
- class [Rcpp::sugar::Divides\\_Vector\\_Vector](#)< RTYPE, false, LHS\_T, RHS\_NA, RHS\_T >
- class [Rcpp::sugar::Divides\\_Vector\\_Vector](#)< REALSXP, false, LHS\_T, RHS\_NA, RHS\_T >
- class [Rcpp::sugar::Divides\\_Vector\\_Vector](#)< RTYPE, LHS\_NA, LHS\_T, false, RHS\_T >
- class [Rcpp::sugar::Divides\\_Vector\\_Vector](#)< REALSXP, LHS\_NA, LHS\_T, false, RHS\_T >
- class [Rcpp::sugar::Divides\\_Vector\\_Vector](#)< RTYPE, false, LHS\_T, false, RHS\_T >
- class [Rcpp::sugar::Divides\\_Vector\\_Vector](#)< REALSXP, false, LHS\_T, false, RHS\_T >
- class [Rcpp::sugar::Divides\\_Vector\\_Primitive](#)< RTYPE, NA, T >
- class [Rcpp::sugar::Divides\\_Vector\\_Primitive](#)< REALSXP, NA, T >
- class [Rcpp::sugar::Divides\\_Vector\\_Primitive](#)< RTYPE, false, T >
- class [Rcpp::sugar::Divides\\_Vector\\_Primitive](#)< REALSXP, false, T >
- class [Rcpp::sugar::Divides\\_Primitive\\_Vector](#)< RTYPE, NA, T >
- class [Rcpp::sugar::Divides\\_Primitive\\_Vector](#)< REALSXP, NA, T >
- class [Rcpp::sugar::Divides\\_Primitive\\_Vector](#)< RTYPE, false, T >
- class [Rcpp::sugar::Divides\\_Primitive\\_Vector](#)< REALSXP, false, T >

## Namespaces

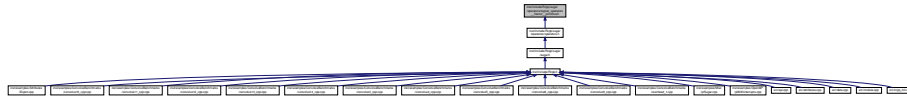
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T, typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename`  
`traits::storage_type< RTYPE >::type >::value, sugar::Divides_Vector_Primitive< RTYPE, NA, T > >::type`  
[Rcpp::operator/](#) (const VectorBase< RTYPE, NA, T > &lhs, const U &rhs)
- `template<int RTYPE, bool NA, typename T, typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename`  
`traits::storage_type< RTYPE >::type >::value, sugar::Divides_Primitive_Vector< RTYPE, NA, T > >::type`  
[Rcpp::operator/](#) (const U &lhs, const VectorBase< RTYPE, NA, T > &rhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`sugar::Divides_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >` [Rcpp::operator/](#) (const Vector←  
Base< RTYPE, LHS\_NA, LHS\_T > &lhs, const VectorBase< RTYPE, RHS\_NA, RHS\_T > &rhs)

## 7.315 inst/include/Rcpp/sugar/operators/logical\_operators\_\_Vector\_\_primitive.h File Reference

This graph shows which files directly or indirectly include this file:



## Functions

- `template<int RTYPE, bool NA, typename T >`  
[Rcpp::sugar::Comparator\\_With\\_One\\_Value](#)< RTYPE, Rcpp::sugar::less< RTYPE >, NA, T > [operator<](#) (const  
[Rcpp::VectorBase](#)< RTYPE, NA, T > &lhs, typename [Rcpp::traits::storage\\_type](#)< RTYPE >::type rhs)
- `template<int RTYPE, bool NA, typename T >`  
[Rcpp::sugar::Comparator\\_With\\_One\\_Value](#)< RTYPE, Rcpp::sugar::less< RTYPE >, NA, T > [operator>](#) (type-  
name [Rcpp::traits::storage\\_type](#)< RTYPE >::type rhs, const [Rcpp::VectorBase](#)< RTYPE, NA, T > &lhs)
- `template<int RTYPE, bool NA, typename T >`  
[Rcpp::sugar::Comparator\\_With\\_One\\_Value](#)< RTYPE, Rcpp::sugar::greater< RTYPE >, NA, T > [operator>](#)  
(const [Rcpp::VectorBase](#)< RTYPE, NA, T > &lhs, typename [Rcpp::traits::storage\\_type](#)< RTYPE >::type rhs)
- `template<int RTYPE, bool NA, typename T >`  
[Rcpp::sugar::Comparator\\_With\\_One\\_Value](#)< RTYPE, Rcpp::sugar::greater< RTYPE >, NA, T > [operator<](#)  
(typename [Rcpp::traits::storage\\_type](#)< RTYPE >::type rhs, const [Rcpp::VectorBase](#)< RTYPE, NA, T > &lhs)
- `template<int RTYPE, bool NA, typename T >`  
[Rcpp::sugar::Comparator\\_With\\_One\\_Value](#)< RTYPE, Rcpp::sugar::less\_or\_equal< RTYPE >, NA, T >  
[operator<=](#) (const [Rcpp::VectorBase](#)< RTYPE, NA, T > &lhs, typename [Rcpp::traits::storage\\_type](#)< RTYPE  
>::type rhs)
- `template<int RTYPE, bool NA, typename T >`  
[Rcpp::sugar::Comparator\\_With\\_One\\_Value](#)< RTYPE, Rcpp::sugar::less\_or\_equal< RTYPE >, NA, T >  
[operator>=](#) (typename [Rcpp::traits::storage\\_type](#)< RTYPE >::type rhs, const [Rcpp::VectorBase](#)< RTYPE,  
NA, T > &lhs)

- `template<int RTYPE, bool NA, typename T >`  
`Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::greater_or_equal< RTYPE >, NA, T >`  
`operator>=` (const `Rcpp::VectorBase< RTYPE, NA, T >` &lhs, typename `Rcpp::traits::storage_type< RTYPE >::type` rhs)
- `template<int RTYPE, bool NA, typename T >`  
`Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::greater_or_equal< RTYPE >, NA, T >`  
`operator<=` (typename `Rcpp::traits::storage_type< RTYPE >::type` rhs, const `Rcpp::VectorBase< RTYPE, NA, T >` &lhs)
- `template<int RTYPE, bool NA, typename T >`  
`Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::equal< RTYPE >, NA, T >`  
`operator==` (const `Rcpp::VectorBase< RTYPE, NA, T >` &lhs, typename `Rcpp::traits::storage_type< RTYPE >::type` rhs)
- `template<int RTYPE, bool NA, typename T >`  
`Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::equal< RTYPE >, NA, T >`  
`operator==` (typename `Rcpp::traits::storage_type< RTYPE >::type` rhs, const `Rcpp::VectorBase< RTYPE, NA, T >` &lhs)
- `template<int RTYPE, bool NA, typename T >`  
`Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::not_equal< RTYPE >, NA, T >`  
`operator!=` (const `Rcpp::VectorBase< RTYPE, NA, T >` &lhs, typename `Rcpp::traits::storage_type< RTYPE >::type` rhs)
- `template<int RTYPE, bool NA, typename T >`  
`Rcpp::sugar::Comparator_With_One_Value< RTYPE, Rcpp::sugar::not_equal< RTYPE >, NA, T >`  
`operator!=` (typename `Rcpp::traits::storage_type< RTYPE >::type` rhs, const `Rcpp::VectorBase< RTYPE, NA, T >` &lhs)

## 7.315.1 Function Documentation

### 7.315.1.1 `operator"!=()` [1/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::not_equal<RTYPE>, NA, T > operator!=
(
    const Rcpp::VectorBase< RTYPE, NA, T > & lhs,
    typename Rcpp::traits::storage_type< RTYPE >::type rhs ) [inline]
```

Definition at line 255 of file `logical_operators__Vector__primitive.h`.

References `Rcpp::NA`.

### 7.315.1.2 `operator"!=()` [2/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::not_equal<RTYPE>, NA, T > operator!=
(
    typename Rcpp::traits::storage_type< RTYPE >::type rhs,
    const Rcpp::VectorBase< RTYPE, NA, T > & lhs ) [inline]
```

Definition at line 275 of file `logical_operators__Vector__primitive.h`.

References `Rcpp::NA`.

### 7.315.1.3 operator<() [1/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::less<RTYPE>, NA, T > operator< (
    const Rcpp::VectorBase< RTYPE, NA, T > & lhs,
    typename Rcpp::traits::storage_type< RTYPE >::type rhs ) [inline]
```

Definition at line 1 of file logical\_operators\_\_Vector\_\_primitive.h.

References Rcpp::NA.

### 7.315.1.4 operator<() [2/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::greater<RTYPE>, NA, T > operator< (
    typename Rcpp::traits::storage_type< RTYPE >::type rhs,
    const Rcpp::VectorBase< RTYPE, NA, T > & lhs ) [inline]
```

Definition at line 78 of file logical\_operators\_\_Vector\_\_primitive.h.

References Rcpp::NA.

### 7.315.1.5 operator<=() [1/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::less_or_equal<RTYPE>, NA, T > operator<=
(
    const Rcpp::VectorBase< RTYPE, NA, T > & lhs,
    typename Rcpp::traits::storage_type< RTYPE >::type rhs ) [inline]
```

Definition at line 78 of file logical\_operators\_\_Vector\_\_primitive.h.

### 7.315.1.6 operator<=() [2/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::greater_or_equal<RTYPE>, NA, T >
operator<= (
    typename Rcpp::traits::storage_type< RTYPE >::type rhs,
    const Rcpp::VectorBase< RTYPE, NA, T > & lhs ) [inline]
```

Definition at line 167 of file logical\_operators\_\_Vector\_\_primitive.h.

References Rcpp::NA.

**7.315.1.7 operator==( ) [1/2]**

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::equal<RTYPE>, NA, T > operator==(
    const Rcpp::VectorBase< RTYPE, NA, T > & lhs,
    typename Rcpp::traits::storage_type< RTYPE >::type rhs ) [inline]
```

Definition at line 211 of file logical\_operators\_\_Vector\_\_primitive.h.

References Rcpp::NA.

**7.315.1.8 operator==( ) [2/2]**

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::equal<RTYPE>, NA, T > operator==(
    typename Rcpp::traits::storage_type< RTYPE >::type rhs,
    const Rcpp::VectorBase< RTYPE, NA, T > & lhs ) [inline]
```

Definition at line 231 of file logical\_operators\_\_Vector\_\_primitive.h.

References Rcpp::NA.

**7.315.1.9 operator>( ) [1/2]**

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::greater<RTYPE>, NA, T > operator> (
    const Rcpp::VectorBase< RTYPE, NA, T > & lhs,
    typename Rcpp::traits::storage_type< RTYPE >::type rhs ) [inline]
```

Definition at line 78 of file logical\_operators\_\_Vector\_\_primitive.h.

**7.315.1.10 operator>( ) [2/2]**

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::less<RTYPE>, NA, T > operator> (
    typename Rcpp::traits::storage_type< RTYPE >::type rhs,
    const Rcpp::VectorBase< RTYPE, NA, T > & lhs ) [inline]
```

Definition at line 54 of file logical\_operators\_\_Vector\_\_primitive.h.

References Rcpp::NA.



**7.315.1.11 operator>=()** [1/2]

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::greater_or_equal<RTYPE>, NA, T >
operator>= (
    const Rcpp::VectorBase< RTYPE, NA, T > & lhs,
    typename Rcpp::traits::storage_type< RTYPE >::type rhs ) [inline]
```

Definition at line 167 of file logical\_operators\_\_Vector\_\_primitive.h.

**7.315.1.12 operator>=()** [2/2]

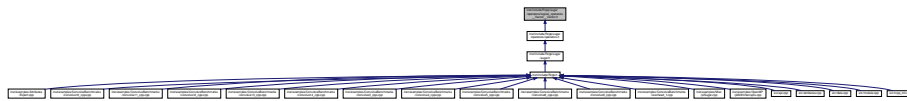
```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::Comparator_With_One_Value< RTYPE , Rcpp::sugar::less_or_equal<RTYPE>, NA, T > operator>=
(
    typename Rcpp::traits::storage_type< RTYPE >::type rhs,
    const Rcpp::VectorBase< RTYPE, NA, T > & lhs ) [inline]
```

Definition at line 142 of file logical\_operators\_\_Vector\_\_primitive.h.

References Rcpp::NA.

## 7.316 inst/include/Rcpp/sugar/operators/logical\_operators\_\_Vector\_\_Vector.h File Reference

This graph shows which files directly or indirectly include this file:



### Functions

- template<int RTYPE, bool LHS\_NA, typename LHS\_T , bool RHS\_NA, typename RHS\_T >  
[Rcpp::sugar::Comparator](#)< RTYPE, Rcpp::sugar::less< RTYPE >, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >  
[operator<](#) (const [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, const [Rcpp::VectorBase](#)< RTYPE,  
RHS\_NA, RHS\_T > &rhs)
- template<int RTYPE, bool LHS\_NA, typename LHS\_T , bool RHS\_NA, typename RHS\_T >  
[Rcpp::sugar::Comparator](#)< RTYPE, Rcpp::sugar::greater< RTYPE >, LHS\_NA, LHS\_T, RHS\_NA, RHS\_T >  
[operator>](#) (const [Rcpp::VectorBase](#)< RTYPE, LHS\_NA, LHS\_T > &lhs, const [Rcpp::VectorBase](#)< RTYPE,  
RHS\_NA, RHS\_T > &rhs)

- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`Rcpp::sugar::Comparator< RTYPE, Rcpp::sugar::less_or_equal< RTYPE >, LHS_NA, LHS_T, RHS_NA,`  
`RHS_T > operator<=` (const `Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T >` &lhs, const `Rcpp::VectorBase<`  
`RTYPE, RHS_NA, RHS_T >` &rhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`Rcpp::sugar::Comparator< RTYPE, Rcpp::sugar::greater_or_equal< RTYPE >, LHS_NA, LHS_T, RHS_NA,`  
`RHS_T > operator>=` (const `Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T >` &lhs, const `Rcpp::VectorBase<`  
`RTYPE, RHS_NA, RHS_T >` &rhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`Rcpp::sugar::Comparator< RTYPE, Rcpp::sugar::equal< RTYPE >, LHS_NA, LHS_T, RHS_NA, RHS_T >`  
`operator==` (const `Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T >` &lhs, const `Rcpp::VectorBase<`  
`RTYPE, RHS_NA, RHS_T >` &rhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`Rcpp::sugar::Comparator< RTYPE, Rcpp::sugar::not_equal< RTYPE >, LHS_NA, LHS_T, RHS_NA, RHS_←`  
`T > operator!=` (const `Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T >` &lhs, const `Rcpp::VectorBase<`  
`RTYPE, RHS_NA, RHS_T >` &rhs)

## 7.316.1 Function Documentation

### 7.316.1.1 operator"!=()

```
template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE, Rcpp::sugar::not_equal<RTYPE>, LHS_NA, LHS_T, RHS_NA, RHS_T >
operator!= (
    const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 139 of file `logical_operators__Vector__Vector.h`.

### 7.316.1.2 operator<()

```
template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE, Rcpp::sugar::less<RTYPE>, LHS_NA, LHS_T, RHS_NA, RHS_T > operator<
(
    const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 1 of file `logical_operators__Vector__Vector.h`.

### 7.316.1.3 operator<=()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE , Rcpp::sugar::less_or_equal<RTYPE>, LHS_NA, LHS_T, RHS_NA, RHS_T
> operator<= (
    const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 55 of file logical\_operators\_\_Vector\_\_Vector.h.

### 7.316.1.4 operator==(())

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE , Rcpp::sugar::equal<RTYPE>, LHS_NA, LHS_T, RHS_NA, RHS_T > operator==(
    const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 118 of file logical\_operators\_\_Vector\_\_Vector.h.

### 7.316.1.5 operator>()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE , Rcpp::sugar::greater<RTYPE>, LHS_NA, LHS_T, RHS_NA, RHS_T >
operator> (
    const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 55 of file logical\_operators\_\_Vector\_\_Vector.h.

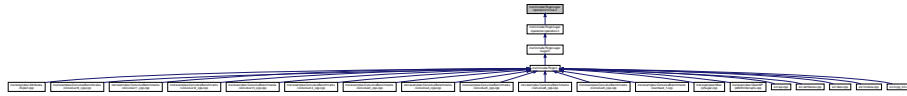
### 7.316.1.6 operator>=()

```
template<int RTYPE, bool LHS_NA, typename LHS_T , bool RHS_NA, typename RHS_T >
Rcpp::sugar::Comparator< RTYPE , Rcpp::sugar::greater_or_equal<RTYPE>, LHS_NA, LHS_T, RHS_NA,
RHS_T > operator>= (
    const Rcpp::VectorBase< RTYPE, LHS_NA, LHS_T > & lhs,
    const Rcpp::VectorBase< RTYPE, RHS_NA, RHS_T > & rhs ) [inline]
```

Definition at line 97 of file logical\_operators\_\_Vector\_\_Vector.h.

## 7.317 inst/include/Rcpp/sugar/operators/minus.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Minus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Minus\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Minus\\_Vector\\_Vector< RTYPE, false, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Minus\\_Vector\\_Vector< REALSXP, false, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Minus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Minus\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Minus\\_Vector\\_Vector< RTYPE, false, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Minus\\_Vector\\_Vector< REALSXP, false, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Minus\\_Vector\\_Primitive< RTYPE, NA, T >](#)
- class [Rcpp::sugar::Minus\\_Vector\\_Primitive< REALSXP, NA, T >](#)
- class [Rcpp::sugar::Minus\\_Vector\\_Primitive< RTYPE, false, T >](#)
- class [Rcpp::sugar::Minus\\_Vector\\_Primitive< REALSXP, false, T >](#)
- class [Rcpp::sugar::Minus\\_Primitive\\_Vector< RTYPE, NA, T >](#)
- class [Rcpp::sugar::Minus\\_Primitive\\_Vector< REALSXP, NA, T >](#)
- class [Rcpp::sugar::Minus\\_Primitive\\_Vector< RTYPE, false, T >](#)
- class [Rcpp::sugar::Minus\\_Primitive\\_Vector< REALSXP, false, T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- `template<int RTYPE, bool NA, typename T, typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Minus_Vector_Primitive< RTYPE, NA, T > >::type`  
[Rcpp::operator-](#) (const VectorBase< RTYPE, NA, T > &lhs, const U &rhs)
- `template<int RTYPE, bool NA, typename T, typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename traits::storage_type< RTYPE >::type >::value, sugar::Minus_Primitive_Vector< RTYPE, NA, T > >::type`  
[Rcpp::operator-](#) (const U &lhs, const VectorBase< RTYPE, NA, T > &rhs)
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`sugar::Minus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T >` [Rcpp::operator-](#) (const VectorBase< RTYPE, LHS\_NA, LHS\_T > &lhs, const VectorBase< RTYPE, RHS\_NA, RHS\_T > &rhs)

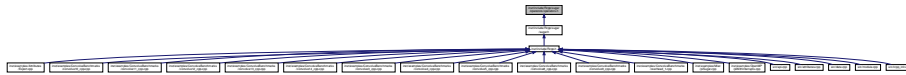
## 7.318 inst/include/Rcpp/sugar/operators/operators.h File Reference

```
#include <Rcpp/sugar/operators/Comparator.h>
#include <Rcpp/sugar/operators/Comparator_With_One_Value.h>
#include <Rcpp/sugar/operators/logical_operators__Vector__Vector.h>
#include <Rcpp/sugar/operators/logical_operators__Vector__primitive.h>
#include <Rcpp/sugar/operators/plus.h>
#include <Rcpp/sugar/operators/minus.h>
#include <Rcpp/sugar/operators/times.h>
#include <Rcpp/sugar/operators/divides.h>
#include <Rcpp/sugar/operators/not.h>
#include <Rcpp/sugar/operators/unary_minus.h>
```

Include dependency graph for operators.h:

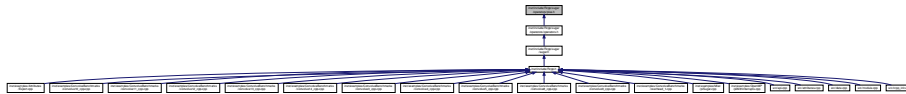


This graph shows which files directly or indirectly include this file:



## 7.319 inst/include/Rcpp/sugar/operators/plus.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::Plus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Vector< RTYPE, false, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Vector< REALSXP, false, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Vector< RTYPE, false, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Vector< REALSXP, false, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Primitive< RTYPE, NA, T >](#)

- class [Rcpp::sugar::Plus\\_Vector\\_Primitive< REALSXP, NA, T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Primitive< RTYPE, false, T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Primitive< REALSXP, false, T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Primitive\\_nona< RTYPE, NA, T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Primitive\\_nona< REALSXP, NA, T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Primitive\\_nona< RTYPE, false, T >](#)
- class [Rcpp::sugar::Plus\\_Vector\\_Primitive\\_nona< REALSXP, false, T >](#)

## Namespaces

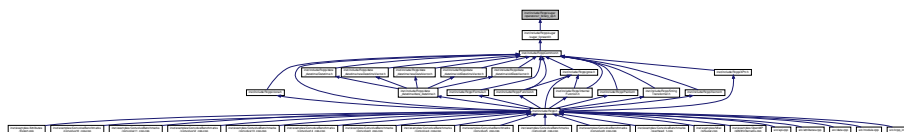
- [Rcpp](#)
  - [Rcpp API](#)
- [Rcpp::sugar](#)

## Functions

- `template<int RTYPE, bool NA, typename T, typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename`  
`traits::storage_type< RTYPE >::type >::value, typename sugar::Plus_Vector_Primitive< RTYPE, NA, T > >←`  
`::type Rcpp::operator+ (const VectorBase< RTYPE, NA, T > &lhs, const U &rhs)`
- `template<int RTYPE, bool NA, typename T, typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename`  
`traits::storage_type< RTYPE >::type >::value, typename sugar::Plus_Vector_Primitive< RTYPE, NA, T > >←`  
`::type Rcpp::operator+ (const U &rhs, const VectorBase< RTYPE, NA, T > &lhs)`
- `template<int RTYPE, bool NA, typename T, typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename`  
`traits::storage_type< RTYPE >::type >::value, sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T > >::type`  
`Rcpp::operator+ (const VectorBase< RTYPE, NA, T > &lhs, const typename sugar::NonaPrimitive< U > &rhs)`
- `template<int RTYPE, bool NA, typename T, typename U >`  
`traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< U >::type, typename`  
`traits::storage_type< RTYPE >::type >::value, sugar::Plus_Vector_Primitive_nona< RTYPE, NA, T > >::type`  
`Rcpp::operator+ (const typename sugar::NonaPrimitive< U > &rhs, const VectorBase< RTYPE, NA, T > &lhs)`
- `template<int RTYPE, bool LHS_NA, typename LHS_T, bool RHS_NA, typename RHS_T >`  
`sugar::Plus_Vector_Vector< RTYPE, LHS_NA, LHS_T, RHS_NA, RHS_T > Rcpp::operator+ (const Vector←`  
`Base< RTYPE, LHS_NA, LHS_T > &lhs, const VectorBase< RTYPE, RHS_NA, RHS_T > &rhs)`

## 7.320 inst/include/Rcpp/sugar/operators/r\_binary\_op.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)
  - Rcpp API.*
- [Rcpp::sugar](#)

## Macros

- #define [RCPP\\_OP](#)(NAME, OP)

### 7.320.1 Macro Definition Documentation

#### 7.320.1.1 RCPP\_OP

```
#define RCPP_OP(
    NAME,
    OP )
```

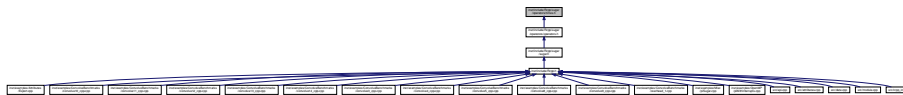
#### Value:

```
template <int RTYPE>
class NAME {
public:
    typedef typename traits::storage_type<RTYPE>::type STORAGE ;
    inline int operator()( STORAGE lhs, STORAGE rhs) const {
        return lhs OP rhs ;
    }
};
```

Definition at line 29 of file r\_binary\_op.h.

## 7.321 inst/include/Rcpp/sugar/operators/times.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::sugar::Times\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Vector< RTYPE, false, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Vector< REALSXP, false, LHS\\_T, RHS\\_NA, RHS\\_T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Vector< REALSXP, LHS\\_NA, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Vector< RTYPE, false, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Vector< REALSXP, false, LHS\\_T, false, RHS\\_T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Primitive< RTYPE, NA, T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Primitive< REALSXP, NA, T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Primitive< RTYPE, false, T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Primitive< REALSXP, false, T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Primitive\\_nona< RTYPE, NA, T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Primitive\\_nona< REALSXP, NA, T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Primitive\\_nona< RTYPE, false, T >](#)
- class [Rcpp::sugar::Times\\_Vector\\_Primitive\\_nona< REALSXP, false, T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

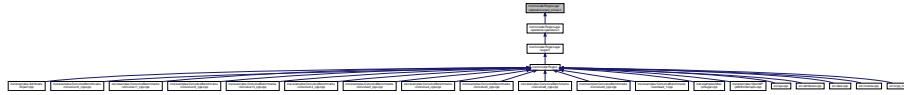
## Functions

- [template<int RTYPE, bool NA, typename T, typename U >](#)  
[traits::enable\\_if< traits::is\\_convertible< typename traits::remove\\_const\\_and\\_reference< U >::type, typename traits::storage\\_type< RTYPE >::type >::value, sugar::Times\\_Vector\\_Primitive< RTYPE, NA, T > >::type](#)  
[Rcpp::operator\\*](#) (const VectorBase< RTYPE, NA, T > &lhs, const U &rhs)
- [template<int RTYPE, bool NA, typename T, typename U >](#)  
[traits::enable\\_if< traits::is\\_convertible< typename traits::remove\\_const\\_and\\_reference< U >::type, typename traits::storage\\_type< RTYPE >::type >::value, sugar::Times\\_Vector\\_Primitive< RTYPE, NA, T > >::type](#)  
[Rcpp::operator\\*](#) (const U &rhs, const VectorBase< RTYPE, NA, T > &lhs)
- [template<int RTYPE, bool NA, typename T, typename U >](#)  
[traits::enable\\_if< traits::is\\_convertible< typename traits::remove\\_const\\_and\\_reference< U >::type, typename traits::storage\\_type< RTYPE >::type >::value, sugar::Times\\_Vector\\_Primitive\\_nona< RTYPE, NA, T > >::type](#)  
[Rcpp::operator\\*](#) (const VectorBase< RTYPE, NA, T > &lhs, const typename sugar::NonaPrimitive< U > &rhs)
- [template<int RTYPE, bool NA, typename T, typename U >](#)  
[traits::enable\\_if< traits::is\\_convertible< typename traits::remove\\_const\\_and\\_reference< U >::type, typename traits::storage\\_type< RTYPE >::type >::value, sugar::Times\\_Vector\\_Primitive\\_nona< RTYPE, NA, T > >::type](#)  
[Rcpp::operator\\*](#) (const typename sugar::NonaPrimitive< U > &rhs, const VectorBase< RTYPE, NA, T > &lhs)
- [template<int RTYPE, bool LHS\\_NA, typename LHS\\_T, bool RHS\\_NA, typename RHS\\_T >](#)  
[sugar::Times\\_Vector\\_Vector< RTYPE, LHS\\_NA, LHS\\_T, RHS\\_NA, RHS\\_T >](#) [Rcpp::operator\\*](#) (const VectorBase< RTYPE, LHS\_NA, LHS\_T > &lhs, const VectorBase< RTYPE, RHS\_NA, RHS\_T > &rhs)



## 7.322 inst/include/Rcpp/sugar/operators/unary\_minus.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::sugar::unary\\_minus\\_result\\_type< RTYPE >](#)
- struct [Rcpp::sugar::unary\\_minus\\_result\\_type< LGLSXP >](#)
- class [Rcpp::sugar::unary\\_minus< RTYPE, NA >](#)
- class [Rcpp::sugar::unary\\_minus< RTYPE, false >](#)
- class [Rcpp::sugar::unary\\_minus< CPLXSXP, NA >](#)
- class [Rcpp::sugar::unary\\_minus< CPLXSXP, false >](#)
- class [Rcpp::sugar::UnaryMinus\\_Vector< RTYPE, NA, T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- [template<int RTYPE, bool NA, typename T > Rcpp::sugar::UnaryMinus\\_Vector< RTYPE, NA, T > operator-](#) (const [Rcpp::VectorBase< RTYPE, NA, T >](#) &x)

### 7.322.1 Function Documentation

#### 7.322.1.1 operator-()

```
template<int RTYPE, bool NA, typename T >
Rcpp::sugar::UnaryMinus_Vector< RTYPE , NA , T > operator- (
    const Rcpp::VectorBase< RTYPE, NA, T > & x ) [inline]
```

Definition at line 113 of file unary\_minus.h.

## 7.323 inst/include/Rcpp/sugar/Range.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::Range](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.324 inst/include/Rcpp/sugar/sets.h File Reference

### Macros

- #define [RCPP\\_UNORDERED\\_SET](#) `std::set`
- #define [RCPP\\_UNORDERED\\_MAP](#) `std::map`

### 7.324.1 Macro Definition Documentation

#### 7.324.1.1 RCPP\_UNORDERED\_MAP

```
#define RCPP_UNORDERED_MAP std::map
```

Definition at line 58 of file sets.h.

#### 7.324.1.2 RCPP\_UNORDERED\_SET

```
#define RCPP_UNORDERED_SET std::set
```

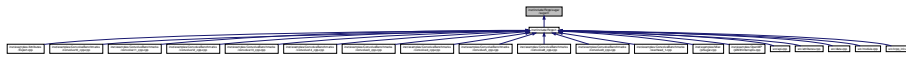
Definition at line 57 of file sets.h.

## 7.325 inst/include/Rcpp/sugar/sugar.h File Reference

```
#include <Rcpp/sugar/tools/iterator.h>
#include <Rcpp/sugar/block/block.h>
#include <Rcpp/hash/hash.h>
#include <Rcpp/sugar/operators/operators.h>
#include <Rcpp/sugar/functions/functions.h>
#include <Rcpp/sugar/matrix/matrix_functions.h>
Include dependency graph for sugar.h:
```

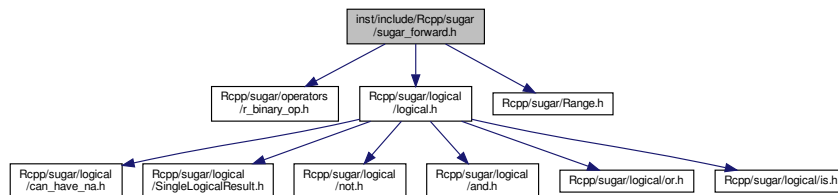


This graph shows which files directly or indirectly include this file:



## 7.326 inst/include/Rcpp/sugar/sugar\_forward.h File Reference

```
#include <Rcpp/sugar/operators/r_binary_op.h>
#include <Rcpp/sugar/logical/logical.h>
#include <Rcpp/sugar/Range.h>
Include dependency graph for sugar_forward.h:
```

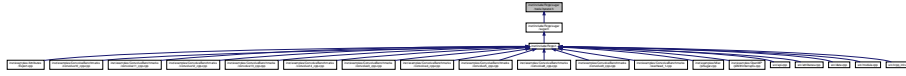


This graph shows which files directly or indirectly include this file:



## 7.327 inst/include/Rcpp/sugar/tools/iterator.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::sugar::SugarIterator< T >](#)
- struct [Rcpp::sugar::sugar\\_const\\_iterator\\_type< T >](#)
- struct [Rcpp::sugar::sugar\\_const\\_iterator\\_type< Rcpp::Vector< RTYPE > >](#)
- struct [Rcpp::sugar::sugar\\_const\\_iterator\\_type< CharacterVector >](#)
- struct [Rcpp::sugar::is\\_sugar\\_vector< T >](#)
- struct [Rcpp::sugar::is\\_sugar\\_vector< Rcpp::Vector< RTYPE > >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- [template<typename T > sugar\\_const\\_iterator\\_type< T >::type Rcpp::sugar::get\\_const\\_begin\\_\\_impl \(const T &obj, Rcpp::traits::true\\_type\)](#)
- [template<typename T > sugar\\_const\\_iterator\\_type< T >::type Rcpp::sugar::get\\_const\\_begin\\_\\_impl \(const T &obj, Rcpp::traits::false\\_type\)](#)
- [template<typename T > sugar\\_const\\_iterator\\_type< T >::type Rcpp::sugar::get\\_const\\_begin \(const T &obj\)](#)
- [template<> SEXP \\* Rcpp::sugar::get\\_const\\_begin \(const CharacterVector &obj\)](#)
- [template<typename T > sugar\\_const\\_iterator\\_type< T >::type Rcpp::sugar::get\\_const\\_end \(const T &obj\)](#)

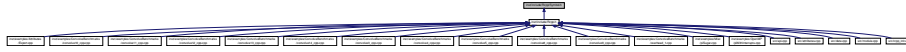
## 7.328 inst/include/Rcpp/sugar/undoRmath.h File Reference

This graph shows which files directly or indirectly include this file:



## 7.329 inst/include/Rcpp/Symbol.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [Rcpp](#)  
*Rcpp API.*

### Typedefs

- typedef `Symbol_Impl< NoProtectStorage >` [Rcpp::Symbol](#)

### Functions

- [Rcpp::RCPP\\_API\\_CLASS](#) (Symbol\_Impl)

## 7.330 inst/include/Rcpp/traits/char\_type.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::traits::char\\_type< T >](#)
- struct [Rcpp::traits::char\\_type< const wchar\\_t \\* >](#)
- struct [Rcpp::traits::char\\_type< const char \\* >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.331 inst/include/Rcpp/traits/enable\_if.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::traits::enable\\_if< B, T >](#)
- struct [Rcpp::traits::enable\\_if< true, T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.332 inst/include/Rcpp/traits/expands\_to\_logical.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::traits::expands\\_to\\_logical\\_\\_impl< RTYPE >](#)
- struct [Rcpp::traits::expands\\_to\\_logical\\_\\_impl< LGSLXP >](#)
- struct [Rcpp::traits::expands\\_to\\_logical\\_\\_impl< LGSLXP >::r\\_expands\\_to\\_logical](#)
- class [Rcpp::traits::\\_has\\_rtype\\_helper< T >](#)
- struct [Rcpp::traits::\\_has\\_rtype\\_helper< T >::\\_Wrap\\_type< U >](#)
- struct [Rcpp::traits::expands\\_to\\_logical< T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.333 inst/include/Rcpp/traits/get\_na.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## Functions

- `template<int RTYPE>`  
`storage_type< RTYPE >::type Rcpp::traits::get_na ()`
- `template<> int Rcpp::traits::get_na< INTSXP > ()`
- `template<> int Rcpp::traits::get_na< LGLSXP > ()`
- `template<> double Rcpp::traits::get_na< REALSXP > ()`
- `template<> Rcomplex Rcpp::traits::get_na< CPLXSXP > ()`
- `template<> SEXP Rcpp::traits::get_na< STRSXP > ()`
- `template<> SEXP Rcpp::traits::get_na< VECSXP > ()`

## 7.334 inst/include/Rcpp/traits/has\_iterator.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::\\_\\_sfinae\\_types](#)
- struct [Rcpp::traits::\\_\\_sfinae\\_types::\\_\\_two](#)
- class [Rcpp::traits::\\_has\\_iterator\\_helper< T >](#)
- struct [Rcpp::traits::\\_has\\_iterator\\_helper< T >::\\_Wrap\\_type< U >](#)
- class [Rcpp::traits::\\_is\\_importer\\_helper< T >](#)
- struct [Rcpp::traits::\\_is\\_importer\\_helper< T >::\\_Wrap\\_type< U >](#)
- class [Rcpp::traits::\\_is\\_generator\\_helper< T >](#)
- struct [Rcpp::traits::\\_is\\_generator\\_helper< T >::\\_Wrap\\_type< U >](#)
- class [Rcpp::traits::\\_is\\_exporter\\_helper< T >](#)
- struct [Rcpp::traits::\\_is\\_exporter\\_helper< T >::\\_Wrap\\_type< U >](#)
- struct [Rcpp::traits::has\\_iterator< T >](#)
- struct [Rcpp::traits::is\\_importer< T >](#)
- struct [Rcpp::traits::is\\_exporter< T >](#)
- struct [Rcpp::traits::is\\_generator< T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.335 inst/include/Rcpp/traits/has\_na.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::has\\_na< RTYPE >](#)
- struct [Rcpp::traits::has\\_na< INTSXP >](#)
- struct [Rcpp::traits::has\\_na< REALSXP >](#)
- struct [Rcpp::traits::has\\_na< CPLXSXP >](#)
- struct [Rcpp::traits::has\\_na< STRSXP >](#)
- struct [Rcpp::traits::has\\_na< LGLSXP >](#)



## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.336 inst/include/Rcpp/traits/if\_.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::if\\_< COND, LHS, RHS >](#)
- struct [Rcpp::traits::if\\_< false, LHS, RHS >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.337 inst/include/Rcpp/traits/init\_type.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::init\\_type< RTYPE >](#)
- struct [Rcpp::traits::init\\_type< STRSXP >](#)
- struct [Rcpp::traits::init\\_type< LGLSXP >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.338 inst/include/Rcpp/traits/integral\_constant.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::integral\\_constant<\\_T, \\_V >](#)
- struct [Rcpp::traits::both<T, U >](#)

## Namespaces

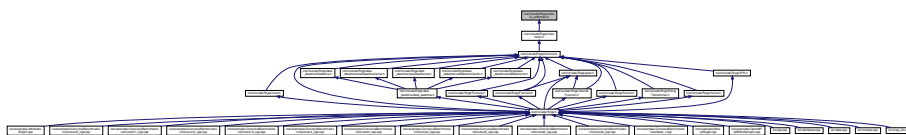
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## Typedefs

- typedef `integral_constant< bool, true >` [Rcpp::traits::true\\_type](#)
- typedef `integral_constant< bool, false >` [Rcpp::traits::false\\_type](#)

## 7.339 inst/include/Rcpp/traits/is\_arithmetic.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::is\\_arithmetic< typename >](#)
- struct [Rcpp::traits::is\\_arithmetic< short >](#)
- struct [Rcpp::traits::is\\_arithmetic< const short >](#)
- struct [Rcpp::traits::is\\_arithmetic< unsigned short >](#)
- struct [Rcpp::traits::is\\_arithmetic< const unsigned short >](#)
- struct [Rcpp::traits::is\\_arithmetic< int >](#)
- struct [Rcpp::traits::is\\_arithmetic< const int >](#)
- struct [Rcpp::traits::is\\_arithmetic< unsigned int >](#)
- struct [Rcpp::traits::is\\_arithmetic< const unsigned int >](#)
- struct [Rcpp::traits::is\\_arithmetic< long >](#)
- struct [Rcpp::traits::is\\_arithmetic< const long >](#)
- struct [Rcpp::traits::is\\_arithmetic< unsigned long >](#)
- struct [Rcpp::traits::is\\_arithmetic< const unsigned long >](#)
- struct [Rcpp::traits::is\\_arithmetic< float >](#)
- struct [Rcpp::traits::is\\_arithmetic< const float >](#)
- struct [Rcpp::traits::is\\_arithmetic< double >](#)
- struct [Rcpp::traits::is\\_arithmetic< const double >](#)
- struct [Rcpp::traits::is\\_arithmetic< long double >](#)
- struct [Rcpp::traits::is\\_arithmetic< const long double >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.340 inst/include/Rcpp/traits/is\_bool.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::is\\_bool< typename >](#)
- struct [Rcpp::traits::is\\_bool< bool >](#)
- struct [Rcpp::traits::is\\_bool< const bool >](#)
- struct [Rcpp::traits::is\\_bool< volatile bool >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

### 7.341 inst/include/Rcpp/traits/is\_const.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::is\\_const< typename >](#)  
*type properties [4.5.3].*
- struct [Rcpp::traits::is\\_const< \\_Tp const >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

### 7.342 inst/include/Rcpp/traits/is\_convertible.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::traits::is\\_convertible< T, U >](#)
- struct [Rcpp::traits::is\\_convertible< T, U >::Big](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.343 inst/include/Rcpp/traits/is\_eigen\_base.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::traits::is\\_eigen\\_helper< T >](#)
- struct [Rcpp::traits::is\\_eigen\\_helper< T >::\\_Wrap\\_type< U >](#)
- struct [Rcpp::traits::is\\_eigen\\_base< T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.344 inst/include/Rcpp/traits/is\_module\_object.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::is\\_module\\_object< T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.345 inst/include/Rcpp/traits/is\_pointer.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::is\\_pointer< T >](#)
- struct [Rcpp::traits::is\\_pointer< T \\* >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.346 inst/include/Rcpp/traits/is\_primitive.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::is\\_primitive< T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.347 inst/include/Rcpp/traits/is\_reference.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::is\\_reference< typename >](#)
- struct [Rcpp::traits::is\\_reference< \\_Tp & >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.348 inst/include/Rcpp/traits/is\_sugar\_expression.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::traits::\\_is\\_sugar\\_expression\\_helper< T >](#)
- struct [Rcpp::traits::\\_is\\_sugar\\_expression\\_helper< T >::\\_Wrap\\_type< U >](#)
- struct [Rcpp::traits::is\\_sugar\\_expression< T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

### 7.349 inst/include/Rcpp/traits/is\_trivial.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::is\\_trivial< RTYPE >](#)
- struct [Rcpp::traits::is\\_trivial< VECSXP >](#)
- struct [Rcpp::traits::is\\_trivial< EXPRSXP >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

### 7.350 inst/include/Rcpp/traits/is\_wide\_string.h File Reference

This graph shows which files directly or indirectly include this file:





## Classes

- struct [Rcpp::traits::is\\_wide\\_string< T >](#)
- struct [Rcpp::traits::is\\_wide\\_string< const wchar\\_t \\* >](#)
- struct [Rcpp::traits::is\\_wide\\_string< const char \\* >](#)
- struct [Rcpp::traits::is\\_wide\\_string< wchar\\_t >](#)
- struct [Rcpp::traits::is\\_wide\\_string< char >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.351 inst/include/Rcpp/traits/matrix\_interface.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::traits::\\_has\\_matrix\\_interface\\_helper< T >](#)
- struct [Rcpp::traits::\\_has\\_matrix\\_interface\\_helper< T >::\\_Wrap\\_type< U >](#)
- struct [Rcpp::traits::matrix\\_interface< T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.352 inst/include/Rcpp/traits/module\_wrap\_traits.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::normal\\_wrap\\_tag](#)
- struct [Rcpp::traits::void\\_wrap\\_tag](#)
- struct [Rcpp::traits::pointer\\_wrap\\_tag](#)
- struct [Rcpp::traits::module\\_wrap\\_traits< T >](#)
- struct [Rcpp::traits::module\\_wrap\\_traits< void >](#)
- struct [Rcpp::traits::module\\_wrap\\_traits< T \\* >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.353 inst/include/Rcpp/traits/named\_object.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::needs\\_protection< T >](#)
- struct [Rcpp::traits::needs\\_protection< SEXP >](#)
- class [Rcpp::traits::named\\_object< T >](#)
- class [Rcpp::traits::named\\_object< SEXP >](#)
- struct [Rcpp::traits::is\\_named< T >](#)
- struct [Rcpp::traits::is\\_named< named\\_object< T > >](#)
- struct [Rcpp::traits::is\\_named< Rcpp::Argument >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.354 inst/include/Rcpp/traits/num2type.h File Reference

### Classes

- struct [Rcpp::traits::num2type< N >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.355 inst/include/Rcpp/traits/one\_type.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

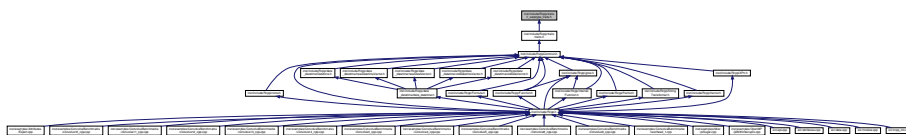
- struct [Rcpp::traits::allowed\\_matrix\\_type< true >](#)
- class [Rcpp::traits::one\\_type< T >](#)
- class [Rcpp::traits::zero\\_type< T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.356 inst/include/Rcpp/traits/r\_sexptype\_traits.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::r\\_sexptype\\_traits< T >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< int >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< const int >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< double >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< const double >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< bool >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< std::string >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< Rcomplex >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< Rbyte >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< unsigned int >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< float >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< long >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< unsigned long >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< long double >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< short >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< unsigned short >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< std::complex< double > >](#)
- struct [Rcpp::traits::r\\_sexptype\\_traits< std::complex< float > >](#)
- struct [Rcpp::traits::r\\_sexptype\\_needs cast< T >](#)
- struct [Rcpp::traits::r\\_sexptype\\_needs cast< int >](#)
- struct [Rcpp::traits::r\\_sexptype\\_needs cast< double >](#)
- struct [Rcpp::traits::r\\_sexptype\\_needs cast< Rcomplex >](#)
- struct [Rcpp::traits::r\\_sexptype\\_needs cast< Rbyte >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.357 inst/include/Rcpp/traits/r\_type\_traits.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::r\\_type\\_primitive\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_string\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_generic\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_pairstring\\_primitive\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_pairstring\\_string\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_pairstring\\_generic\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_pair\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_module\\_object\\_pointer\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_module\\_object\\_const\\_pointer\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_module\\_object\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_module\\_object\\_reference\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_module\\_object\\_const\\_reference\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_enum\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_RcppString\\_tag](#)
- struct [Rcpp::traits::r\\_type\\_traits< T >](#)
- struct [Rcpp::traits::r\\_type\\_traits< Rcpp::object< T > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const KEY, VALUE > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, T > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, int > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, const int > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, double > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, Rbyte > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, Rcomplex > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, bool > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, std::string > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, std::wstring > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, char > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, wchar\\_t > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, unsigned int > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, float > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< int >](#)
- struct [Rcpp::traits::r\\_type\\_traits< const int >](#)
- struct [Rcpp::traits::r\\_type\\_traits< double >](#)
- struct [Rcpp::traits::r\\_type\\_traits< const double >](#)
- struct [Rcpp::traits::r\\_type\\_traits< Rbyte >](#)
- struct [Rcpp::traits::r\\_type\\_traits< Rcomplex >](#)
- struct [Rcpp::traits::r\\_type\\_traits< bool >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::string >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::wstring >](#)
- struct [Rcpp::traits::r\\_type\\_traits< char >](#)
- struct [Rcpp::traits::r\\_type\\_traits< wchar\\_t >](#)
- struct [Rcpp::traits::r\\_type\\_traits< unsigned int >](#)
- struct [Rcpp::traits::r\\_type\\_traits< float >](#)
- struct [Rcpp::traits::r\\_type\\_traits< const char \\* >](#)
- struct [Rcpp::traits::r\\_type\\_traits< const wchar\\_t \\* >](#)
- struct [Rcpp::traits::r\\_type\\_traits< long >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, long > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< unsigned long >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, unsigned long > >](#)

- struct [Rcpp::traits::r\\_type\\_traits< long double >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, long double > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< short >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, short > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< unsigned short >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, unsigned short > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::complex< double > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, std::complex< double > > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::complex< float > >](#)
- struct [Rcpp::traits::r\\_type\\_traits< std::pair< const std::string, std::complex< float > > >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.358 inst/include/Rcpp/traits/remove\_const.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::remove\\_const< \\_Tp >](#)  
*const-volatile modifications [4.7.1].*
- struct [Rcpp::traits::remove\\_const< \\_Tp const >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.359 inst/include/Rcpp/traits/remove\_const\_and\_reference.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::traits::remove\\_const\\_and\\_reference< T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.360 inst/include/Rcpp/traits/remove\_reference.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::traits::remove\\_reference< \\_Tp >](#)  
*reference modifications [4.7.2].*
- struct [Rcpp::traits::remove\\_reference< \\_Tp & >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.361 inst/include/Rcpp/traits/result\_of.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::traits::result\\_of< T >](#)
- struct [Rcpp::traits::result\\_of< RESULT\\_TYPE\(\\*\) \(INPUT\\_TYPE\) >](#)
- struct [Rcpp::traits::result\\_of< RESULT\\_TYPE\(\\*\) \(U1, U2\) >](#)
- struct [Rcpp::traits::result\\_of< RESULT\\_TYPE\(\\*\) \(U1, U2, U3\) >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.362 inst/include/Rcpp/traits/same\_type.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::traits::same\\_type< T, U >](#)
- struct [Rcpp::traits::same\\_type< T, T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*



## 7.363 inst/include/Rcpp/traits/storage\_type.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::traits::storage\\_type< RTYPE >](#)
- struct [Rcpp::traits::storage\\_type< INTSXP >](#)
- struct [Rcpp::traits::storage\\_type< REALSXP >](#)
- struct [Rcpp::traits::storage\\_type< CPLXSXP >](#)
- struct [Rcpp::traits::storage\\_type< RAWSXP >](#)
- struct [Rcpp::traits::storage\\_type< LGLSXP >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.364 inst/include/Rcpp/traits/un\_pointer.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- struct [Rcpp::traits::un\\_pointer< T >](#)
- struct [Rcpp::traits::un\\_pointer< T \\* >](#)
- struct [Rcpp::traits::un\\_pointer< object< T > >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.365 inst/include/Rcpp/traits/wrap\_type\_traits.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [Rcpp::traits::wrap\\_type\\_primitive\\_tag](#)
- struct [Rcpp::traits::wrap\\_type\\_unknown\\_tag](#)
- struct [Rcpp::traits::wrap\\_type\\_module\\_object\\_pointer\\_tag](#)
- struct [Rcpp::traits::wrap\\_type\\_module\\_object\\_tag](#)
- struct [Rcpp::traits::wrap\\_type\\_enum\\_tag](#)
- struct [Rcpp::traits::wrap\\_type\\_char\\_array](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< T >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< char\[N\]>](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< const char\[N\]>](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< int >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< const int >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< double >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< Rbyte >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< Rcomplex >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< unsigned int >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< bool >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< std::string >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< std::wstring >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< Rcpp::String >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< char >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< wchar\\_t >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< float >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< std::complex< float > >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< std::complex< double > >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< long >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< unsigned long >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< long double >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< short >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< unsigned short >](#)
- struct [Rcpp::traits::wrap\\_type\\_traits< Rcpp::object< T > >](#)

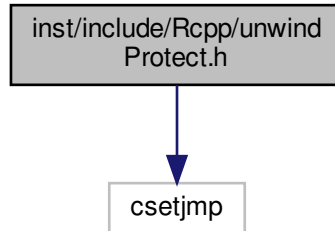
## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.366 inst/include/Rcpp/unwindProtect.h File Reference

```
#include <csetjmp>
```

Include dependency graph for unwindProtect.h:



## Classes

- struct [Rcpp::internal::UnwindData](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

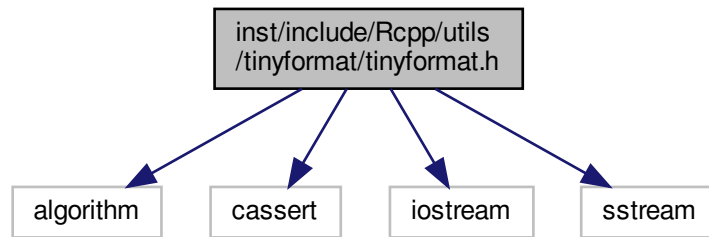
## Functions

- void [Rcpp::internal::maybeJump](#) (void \*unwind\_data, Rboolean jump)
- SEXP [Rcpp::unwindProtect](#) (SEXP(\*callback)(void \*data), void \*data)

## 7.367 inst/include/Rcpp/utis/tinyformat/tinyformat.h File Reference

```
#include <algorithm>
#include <cassert>
#include <iostream>
#include <sstream>
```

Include dependency graph for tinyformat.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct `tinyformat::detail::is_convertible< T1, T2 >`
- struct `tinyformat::detail::is_convertible< T1, T2 >::fail`
- struct `tinyformat::detail::is_convertible< T1, T2 >::succeed`
- struct `tinyformat::detail::is_wchar< T >`
- struct `tinyformat::detail::is_wchar< wchar_t * >`
- struct `tinyformat::detail::is_wchar< const wchar_t * >`
- struct `tinyformat::detail::is_wchar< const wchar_t[n]>`
- struct `tinyformat::detail::is_wchar< wchar_t[n]>`
- struct `tinyformat::detail::formatValueAsType< T, fmtT, convertible >`
- struct `tinyformat::detail::formatValueAsType< T, fmtT, true >`
- struct `tinyformat::detail::convertToInt< T, convertible >`
- struct `tinyformat::detail::convertToInt< T, true >`
- class `tinyformat::detail::FormatArg`
- class `tinyformat::FormatList`
- class `tinyformat::detail::FormatListN< N >`
- class `tinyformat::detail::FormatListN< 0 >`

## Namespaces

- [tinyformat](#)
- [tinyformat::detail](#)

## Macros

- #define [TINYFORMAT\\_ASSERT](#)(cond) assert(cond)
- #define [TINYFORMAT\\_ERROR](#)(reason) assert(0 && reason)
- #define [TINYFORMAT\\_HIDDEN](#)
- #define [TINYFORMAT\\_DEFINE\\_FORMAT\\_TRUNCATED\\_CSTR](#)(type)
- #define [TINYFORMAT\\_DEFINE\\_FORMATVALUE\\_CHAR](#)(charType)
- #define [TINYFORMAT\\_ARGTYPES](#)(n) TINYFORMAT\_ARGTYPES\_ ## n
- #define [TINYFORMAT\\_VARARGS](#)(n) TINYFORMAT\_VARARGS\_ ## n
- #define [TINYFORMAT\\_PASSARGS](#)(n) TINYFORMAT\_PASSARGS\_ ## n
- #define [TINYFORMAT\\_PASSARGS\\_TAIL](#)(n) TINYFORMAT\_PASSARGS\_TAIL\_ ## n
- #define [TINYFORMAT\\_ARGTYPES\\_1](#) class T1
- #define [TINYFORMAT\\_ARGTYPES\\_2](#) class T1, class T2
- #define [TINYFORMAT\\_ARGTYPES\\_3](#) class T1, class T2, class T3
- #define [TINYFORMAT\\_ARGTYPES\\_4](#) class T1, class T2, class T3, class T4
- #define [TINYFORMAT\\_ARGTYPES\\_5](#) class T1, class T2, class T3, class T4, class T5
- #define [TINYFORMAT\\_ARGTYPES\\_6](#) class T1, class T2, class T3, class T4, class T5, class T6
- #define [TINYFORMAT\\_ARGTYPES\\_7](#) class T1, class T2, class T3, class T4, class T5, class T6, class T7
- #define [TINYFORMAT\\_ARGTYPES\\_8](#) class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8
- #define [TINYFORMAT\\_ARGTYPES\\_9](#) class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9
- #define [TINYFORMAT\\_ARGTYPES\\_10](#) class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10
- #define [TINYFORMAT\\_ARGTYPES\\_11](#) class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10, class T11
- #define [TINYFORMAT\\_ARGTYPES\\_12](#) class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10, class T11, class T12
- #define [TINYFORMAT\\_ARGTYPES\\_13](#) class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10, class T11, class T12, class T13
- #define [TINYFORMAT\\_ARGTYPES\\_14](#) class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10, class T11, class T12, class T13, class T14
- #define [TINYFORMAT\\_ARGTYPES\\_15](#) class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10, class T11, class T12, class T13, class T14, class T15
- #define [TINYFORMAT\\_ARGTYPES\\_16](#) class T1, class T2, class T3, class T4, class T5, class T6, class T7, class T8, class T9, class T10, class T11, class T12, class T13, class T14, class T15, class T16
- #define [TINYFORMAT\\_VARARGS\\_1](#) const T1& v1
- #define [TINYFORMAT\\_VARARGS\\_2](#) const T1& v1, const T2& v2
- #define [TINYFORMAT\\_VARARGS\\_3](#) const T1& v1, const T2& v2, const T3& v3
- #define [TINYFORMAT\\_VARARGS\\_4](#) const T1& v1, const T2& v2, const T3& v3, const T4& v4
- #define [TINYFORMAT\\_VARARGS\\_5](#) const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5
- #define [TINYFORMAT\\_VARARGS\\_6](#) const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6
- #define [TINYFORMAT\\_VARARGS\\_7](#) const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7

- #define [TINYFORMAT\\_VARARGS\\_8](#) const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8
- #define [TINYFORMAT\\_VARARGS\\_9](#) const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9
- #define [TINYFORMAT\\_VARARGS\\_10](#) const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10
- #define [TINYFORMAT\\_VARARGS\\_11](#) const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11
- #define [TINYFORMAT\\_VARARGS\\_12](#) const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12
- #define [TINYFORMAT\\_VARARGS\\_13](#) const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12, const T13& v13
- #define [TINYFORMAT\\_VARARGS\\_14](#) const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12, const T13& v13, const T14& v14
- #define [TINYFORMAT\\_VARARGS\\_15](#) const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12, const T13& v13, const T14& v14, const T15& v15
- #define [TINYFORMAT\\_VARARGS\\_16](#) const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12, const T13& v13, const T14& v14, const T15& v15, const T16& v16
- #define [TINYFORMAT\\_PASSARGS\\_1](#) v1
- #define [TINYFORMAT\\_PASSARGS\\_2](#) v1, v2
- #define [TINYFORMAT\\_PASSARGS\\_3](#) v1, v2, v3
- #define [TINYFORMAT\\_PASSARGS\\_4](#) v1, v2, v3, v4
- #define [TINYFORMAT\\_PASSARGS\\_5](#) v1, v2, v3, v4, v5
- #define [TINYFORMAT\\_PASSARGS\\_6](#) v1, v2, v3, v4, v5, v6
- #define [TINYFORMAT\\_PASSARGS\\_7](#) v1, v2, v3, v4, v5, v6, v7
- #define [TINYFORMAT\\_PASSARGS\\_8](#) v1, v2, v3, v4, v5, v6, v7, v8
- #define [TINYFORMAT\\_PASSARGS\\_9](#) v1, v2, v3, v4, v5, v6, v7, v8, v9
- #define [TINYFORMAT\\_PASSARGS\\_10](#) v1, v2, v3, v4, v5, v6, v7, v8, v9, v10
- #define [TINYFORMAT\\_PASSARGS\\_11](#) v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11
- #define [TINYFORMAT\\_PASSARGS\\_12](#) v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12
- #define [TINYFORMAT\\_PASSARGS\\_13](#) v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13
- #define [TINYFORMAT\\_PASSARGS\\_14](#) v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14
- #define [TINYFORMAT\\_PASSARGS\\_15](#) v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15
- #define [TINYFORMAT\\_PASSARGS\\_16](#) v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15, v16
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_1](#)
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_2](#) , v2
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_3](#) , v2, v3
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_4](#) , v2, v3, v4
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_5](#) , v2, v3, v4, v5
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_6](#) , v2, v3, v4, v5, v6
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_7](#) , v2, v3, v4, v5, v6, v7
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_8](#) , v2, v3, v4, v5, v6, v7, v8
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_9](#) , v2, v3, v4, v5, v6, v7, v8, v9
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_10](#) , v2, v3, v4, v5, v6, v7, v8, v9, v10
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_11](#) , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_12](#) , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_13](#) , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13

- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_14](#) , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_15](#) , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15
- #define [TINYFORMAT\\_PASSARGS\\_TAIL\\_16](#) , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15, v16
- #define [TINYFORMAT\\_FOREACH\\_ARGNUM](#)(m) m(1) m(2) m(3) m(4) m(5) m(6) m(7) m(8) m(9) m(10) m(11) m(12) m(13) m(14) m(15) m(16)
- #define [TINYFORMAT\\_MAKE\\_FORMATLIST\\_CONSTRUCTOR](#)(n)
- #define [TINYFORMAT\\_MAKE\\_MAKEFORMATLIST](#)(n)
- #define [TINYFORMAT\\_MAKE\\_FORMAT\\_FUNCS](#)(n)

## Typedefs

- typedef const FormatList & [tinyformat::FormatListRef](#)  
Reference to type-opaque format list for passing to [vformat\(\)](#)

## Functions

- template<typename T >  
void [tinyformat::detail::formatTruncated](#) (std::ostream &out, const T &value, int ntrunc)
- template<typename T >  
void [tinyformat::formatValue](#) (std::ostream &out, const char \*, const char \*fmtEnd, int ntrunc, const T &value)
- int [tinyformat::detail::parseIntAndAdvance](#) (const char \*&c)
- const char \* [tinyformat::detail::printFormatStringLiteral](#) (std::ostream &out, const char \*fmt)
- const char \* [tinyformat::detail::streamStateFromFormat](#) (std::ostream &out, bool &spacePadPositive, int &ntrunc, const char \*fmtStart, const detail::FormatArg \*formatters, int &argIndex, int numFormatters)
- void [tinyformat::detail::formatImpl](#) (std::ostream &out, const char \*fmt, const detail::FormatArg \*formatters, int numFormatters)
- detail::FormatListN< 0 > [tinyformat::makeFormatList](#) ()
- void [tinyformat::vformat](#) (std::ostream &out, const char \*fmt, FormatListRef list)
- void [tinyformat::format](#) (std::ostream &out, const char \*fmt)
- std::string [tinyformat::format](#) (const char \*fmt)
- void [tinyformat::printf](#) (const char \*fmt)
- void [tinyformat::println](#) (const char \*fmt)

## 7.367.1 Macro Definition Documentation

### 7.367.1.1 TINYFORMAT\_ARGTYPES

```
#define TINYFORMAT_ARGTYPES(  
    n ) TINYFORMAT_ARGTYPES_ ## n
```

Definition at line 371 of file tinyformat.h.

### 7.367.1.2 TINYFORMAT\_ARGTYPES\_1

```
#define TINYFORMAT_ARGTYPES_1 class T1
```

Definition at line 412 of file tinyformat.h.

### 7.367.1.3 TINYFORMAT\_ARGTYPES\_10

```
#define TINYFORMAT_ARGTYPES_10 class T1, class T2, class T3, class T4, class T5, class T6, class  
T7, class T8, class T9, class T10
```

Definition at line 421 of file tinyformat.h.

### 7.367.1.4 TINYFORMAT\_ARGTYPES\_11

```
#define TINYFORMAT_ARGTYPES_11 class T1, class T2, class T3, class T4, class T5, class T6, class  
T7, class T8, class T9, class T10, class T11
```

Definition at line 422 of file tinyformat.h.

### 7.367.1.5 TINYFORMAT\_ARGTYPES\_12

```
#define TINYFORMAT_ARGTYPES_12 class T1, class T2, class T3, class T4, class T5, class T6, class  
T7, class T8, class T9, class T10, class T11, class T12
```

Definition at line 423 of file tinyformat.h.

### 7.367.1.6 TINYFORMAT\_ARGTYPES\_13

```
#define TINYFORMAT_ARGTYPES_13 class T1, class T2, class T3, class T4, class T5, class T6, class  
T7, class T8, class T9, class T10, class T11, class T12, class T13
```

Definition at line 424 of file tinyformat.h.



### 7.367.1.7 TINYFORMAT\_ARGTYPES\_14

```
#define TINYFORMAT_ARGTYPES_14 class T1, class T2, class T3, class T4, class T5, class T6, class  
T7, class T8, class T9, class T10, class T11, class T12, class T13, class T14
```

Definition at line 425 of file tinyformat.h.

### 7.367.1.8 TINYFORMAT\_ARGTYPES\_15

```
#define TINYFORMAT_ARGTYPES_15 class T1, class T2, class T3, class T4, class T5, class T6, class  
T7, class T8, class T9, class T10, class T11, class T12, class T13, class T14, class T15
```

Definition at line 426 of file tinyformat.h.

### 7.367.1.9 TINYFORMAT\_ARGTYPES\_16

```
#define TINYFORMAT_ARGTYPES_16 class T1, class T2, class T3, class T4, class T5, class T6, class  
T7, class T8, class T9, class T10, class T11, class T12, class T13, class T14, class T15, class  
T16
```

Definition at line 427 of file tinyformat.h.

### 7.367.1.10 TINYFORMAT\_ARGTYPES\_2

```
#define TINYFORMAT_ARGTYPES_2 class T1, class T2
```

Definition at line 413 of file tinyformat.h.

### 7.367.1.11 TINYFORMAT\_ARGTYPES\_3

```
#define TINYFORMAT_ARGTYPES_3 class T1, class T2, class T3
```

Definition at line 414 of file tinyformat.h.

#### 7.367.1.12 TINYFORMAT\_ARGTYPES\_4

```
#define TINYFORMAT_ARGTYPES_4 class T1, class T2, class T3, class T4
```

Definition at line 415 of file tinyformat.h.

#### 7.367.1.13 TINYFORMAT\_ARGTYPES\_5

```
#define TINYFORMAT_ARGTYPES_5 class T1, class T2, class T3, class T4, class T5
```

Definition at line 416 of file tinyformat.h.

#### 7.367.1.14 TINYFORMAT\_ARGTYPES\_6

```
#define TINYFORMAT_ARGTYPES_6 class T1, class T2, class T3, class T4, class T5, class T6
```

Definition at line 417 of file tinyformat.h.

#### 7.367.1.15 TINYFORMAT\_ARGTYPES\_7

```
#define TINYFORMAT_ARGTYPES_7 class T1, class T2, class T3, class T4, class T5, class T6, class T7
```

Definition at line 418 of file tinyformat.h.

#### 7.367.1.16 TINYFORMAT\_ARGTYPES\_8

```
#define TINYFORMAT_ARGTYPES_8 class T1, class T2, class T3, class T4, class T5, class T6, class  
T7, class T8
```

Definition at line 419 of file tinyformat.h.

**7.367.1.17 TINYFORMAT\_ARGTYPES\_9**

```
#define TINYFORMAT_ARGTYPES_9 class T1, class T2, class T3, class T4, class T5, class T6, class
T7, class T8, class T9
```

Definition at line 420 of file tinyformat.h.

**7.367.1.18 TINYFORMAT\_ASSERT**

```
#define TINYFORMAT_ASSERT(
    cond ) assert (cond)
```

Definition at line 142 of file tinyformat.h.

**7.367.1.19 TINYFORMAT\_DEFINE\_FORMAT\_TRUNCATED\_CSTR**

```
#define TINYFORMAT_DEFINE_FORMAT_TRUNCATED_CSTR(
    type )
```

**Value:**

```
inline void formatTruncated(std::ostream& out, type* value, int ntrunc) \
{
    std::streamsize len = 0;
    while(len < ntrunc && value[len] != 0)
        ++len;
    out.write(value, len);
}
```

Definition at line 278 of file tinyformat.h.

**7.367.1.20 TINYFORMAT\_DEFINE\_FORMATVALUE\_CHAR**

```
#define TINYFORMAT_DEFINE_FORMATVALUE_CHAR(
    charType )
```

**Value:**

```
inline void formatValue(std::ostream& out, const char* /*fmtBegin*/, \
    const char* fmtEnd, int , charType value) \
{
    switch(*(fmtEnd-1))
    {
        case 'u': case 'd': case 'i': case 'o': case 'X': case 'x':
            out << static_cast<int>(value); break;
        default:
            out << value; break;
    }
}
```

Definition at line 347 of file tinyformat.h.

### 7.367.1.21 TINYFORMAT\_ERROR

```
#define TINYFORMAT_ERROR(
    reason ) assert(0 && reason)
```

Definition at line 146 of file tinyformat.h.

### 7.367.1.22 TINYFORMAT\_FOREACH\_ARGNUM

```
#define TINYFORMAT_FOREACH_ARGNUM(
    m ) m(1) m(2) m(3) m(4) m(5) m(6) m(7) m(8) m(9) m(10) m(11) m(12) m(13) m(14)
m(15) m(16)
```

Definition at line 480 of file tinyformat.h.

### 7.367.1.23 TINYFORMAT\_HIDDEN

```
#define TINYFORMAT_HIDDEN
```

Definition at line 166 of file tinyformat.h.

### 7.367.1.24 TINYFORMAT\_MAKE\_FORMAT\_FUNCS

```
#define TINYFORMAT_MAKE_FORMAT_FUNCS (
    n )
```

#### Value:

```
template<TINYFORMAT_ARGTYPES(n)>
void format(std::ostream& out, const char* fmt, TINYFORMAT_VARARGS(n))
{
    vformat(out, fmt, makeFormatList(TINYFORMAT_PASSARGS(n)));
}

template<TINYFORMAT_ARGTYPES(n)>
std::string format(const char* fmt, TINYFORMAT_VARARGS(n))
{
    std::ostringstream oss;
    format(oss, fmt, TINYFORMAT_PASSARGS(n));
    return oss.str();
}

template<TINYFORMAT_ARGTYPES(n)>
void printf(const char* fmt, TINYFORMAT_VARARGS(n))
{
    format(std::cout, fmt, TINYFORMAT_PASSARGS(n));
}

template<TINYFORMAT_ARGTYPES(n)>
void printfn(const char* fmt, TINYFORMAT_VARARGS(n))
{
    format(std::cout, fmt, TINYFORMAT_PASSARGS(n));
    std::cout << '\n';
}
```

Definition at line 1019 of file tinyformat.h.

**7.367.1.25 TINYFORMAT\_MAKE\_FORMATLIST\_CONSTRUCTOR**

```
#define TINYFORMAT_MAKE_FORMATLIST_CONSTRUCTOR(
    n )
```

**Value:**

```
template<TINYFORMAT_ARGTYPES(n)>
FormatListN(TINYFORMAT_VARARGS(n))
    : FormatList(&m_formatterStore[0], n)
{ TINYFORMAT_ASSERT(n == N); init(0, TINYFORMAT_PASSARGS(n)); }

template<TINYFORMAT_ARGTYPES(n)>
void init(int i, TINYFORMAT_VARARGS(n))
{
    m_formatterStore[i] = FormatArg(v1);
    init(i+1 TINYFORMAT_PASSARGS_TAIL(n));
}
```

Definition at line 885 of file tinyformat.h.

**7.367.1.26 TINYFORMAT\_MAKE\_MAKEFORMATLIST**

```
#define TINYFORMAT_MAKE_MAKEFORMATLIST(
    n )
```

**Value:**

```
template<TINYFORMAT_ARGTYPES(n)>
detail::FormatListN<n> makeFormatList(TINYFORMAT_VARARGS(n))
{
    return detail::FormatListN<n>(TINYFORMAT_PASSARGS(n));
}
```

Definition at line 939 of file tinyformat.h.

**7.367.1.27 TINYFORMAT\_PASSARGS**

```
#define TINYFORMAT_PASSARGS(
    n ) TINYFORMAT_PASSARGS_ ## n
```

Definition at line 373 of file tinyformat.h.

**7.367.1.28 TINYFORMAT\_PASSARGS\_1**

```
#define TINYFORMAT_PASSARGS_1 v1
```

Definition at line 446 of file tinyformat.h.

### 7.367.1.29 TINYFORMAT\_PASSARGS\_10

```
#define TINYFORMAT_PASSARGS_10 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10
```

Definition at line 455 of file tinyformat.h.

### 7.367.1.30 TINYFORMAT\_PASSARGS\_11

```
#define TINYFORMAT_PASSARGS_11 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11
```

Definition at line 456 of file tinyformat.h.

### 7.367.1.31 TINYFORMAT\_PASSARGS\_12

```
#define TINYFORMAT_PASSARGS_12 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12
```

Definition at line 457 of file tinyformat.h.

### 7.367.1.32 TINYFORMAT\_PASSARGS\_13

```
#define TINYFORMAT_PASSARGS_13 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13
```

Definition at line 458 of file tinyformat.h.

### 7.367.1.33 TINYFORMAT\_PASSARGS\_14

```
#define TINYFORMAT_PASSARGS_14 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14
```

Definition at line 459 of file tinyformat.h.

### 7.367.1.34 TINYFORMAT\_PASSARGS\_15

```
#define TINYFORMAT_PASSARGS_15 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15
```

Definition at line 460 of file tinyformat.h.

### 7.367.1.35 TINYFORMAT\_PASSARGS\_16

```
#define TINYFORMAT_PASSARGS_16 v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15, v16
```

Definition at line 461 of file tinyformat.h.

### 7.367.1.36 TINYFORMAT\_PASSARGS\_2

```
#define TINYFORMAT_PASSARGS_2 v1, v2
```

Definition at line 447 of file tinyformat.h.

### 7.367.1.37 TINYFORMAT\_PASSARGS\_3

```
#define TINYFORMAT_PASSARGS_3 v1, v2, v3
```

Definition at line 448 of file tinyformat.h.

### 7.367.1.38 TINYFORMAT\_PASSARGS\_4

```
#define TINYFORMAT_PASSARGS_4 v1, v2, v3, v4
```

Definition at line 449 of file tinyformat.h.

### 7.367.1.39 TINYFORMAT\_PASSARGS\_5

```
#define TINYFORMAT_PASSARGS_5 v1, v2, v3, v4, v5
```

Definition at line 450 of file tinyformat.h.

### 7.367.1.40 TINYFORMAT\_PASSARGS\_6

```
#define TINYFORMAT_PASSARGS_6 v1, v2, v3, v4, v5, v6
```

Definition at line 451 of file tinyformat.h.

**7.367.1.41 TINYFORMAT\_PASSARGS\_7**

```
#define TINYFORMAT_PASSARGS_7 v1, v2, v3, v4, v5, v6, v7
```

Definition at line 452 of file tinyformat.h.

**7.367.1.42 TINYFORMAT\_PASSARGS\_8**

```
#define TINYFORMAT_PASSARGS_8 v1, v2, v3, v4, v5, v6, v7, v8
```

Definition at line 453 of file tinyformat.h.

**7.367.1.43 TINYFORMAT\_PASSARGS\_9**

```
#define TINYFORMAT_PASSARGS_9 v1, v2, v3, v4, v5, v6, v7, v8, v9
```

Definition at line 454 of file tinyformat.h.

**7.367.1.44 TINYFORMAT\_PASSARGS\_TAIL**

```
#define TINYFORMAT_PASSARGS_TAIL(  
    n ) TINYFORMAT_PASSARGS_TAIL_ ## n
```

Definition at line 374 of file tinyformat.h.

**7.367.1.45 TINYFORMAT\_PASSARGS\_TAIL\_1**

```
#define TINYFORMAT_PASSARGS_TAIL_1
```

Definition at line 463 of file tinyformat.h.

**7.367.1.46 TINYFORMAT\_PASSARGS\_TAIL\_10**

```
#define TINYFORMAT_PASSARGS_TAIL_10 , v2, v3, v4, v5, v6, v7, v8, v9, v10
```

Definition at line 472 of file tinyformat.h.



**7.367.1.47 TINYFORMAT\_PASSARGS\_TAIL\_11**

```
#define TINYFORMAT_PASSARGS_TAIL_11 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11
```

Definition at line 473 of file tinyformat.h.

**7.367.1.48 TINYFORMAT\_PASSARGS\_TAIL\_12**

```
#define TINYFORMAT_PASSARGS_TAIL_12 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12
```

Definition at line 474 of file tinyformat.h.

**7.367.1.49 TINYFORMAT\_PASSARGS\_TAIL\_13**

```
#define TINYFORMAT_PASSARGS_TAIL_13 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13
```

Definition at line 475 of file tinyformat.h.

**7.367.1.50 TINYFORMAT\_PASSARGS\_TAIL\_14**

```
#define TINYFORMAT_PASSARGS_TAIL_14 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14
```

Definition at line 476 of file tinyformat.h.

**7.367.1.51 TINYFORMAT\_PASSARGS\_TAIL\_15**

```
#define TINYFORMAT_PASSARGS_TAIL_15 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15
```

Definition at line 477 of file tinyformat.h.

**7.367.1.52 TINYFORMAT\_PASSARGS\_TAIL\_16**

```
#define TINYFORMAT_PASSARGS_TAIL_16 , v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15, v16
```

Definition at line 478 of file tinyformat.h.

### 7.367.1.53 TINYFORMAT\_PASSARGS\_TAIL\_2

```
#define TINYFORMAT_PASSARGS_TAIL_2 , v2
```

Definition at line 464 of file tinyformat.h.

### 7.367.1.54 TINYFORMAT\_PASSARGS\_TAIL\_3

```
#define TINYFORMAT_PASSARGS_TAIL_3 , v2, v3
```

Definition at line 465 of file tinyformat.h.

### 7.367.1.55 TINYFORMAT\_PASSARGS\_TAIL\_4

```
#define TINYFORMAT_PASSARGS_TAIL_4 , v2, v3, v4
```

Definition at line 466 of file tinyformat.h.

### 7.367.1.56 TINYFORMAT\_PASSARGS\_TAIL\_5

```
#define TINYFORMAT_PASSARGS_TAIL_5 , v2, v3, v4, v5
```

Definition at line 467 of file tinyformat.h.

### 7.367.1.57 TINYFORMAT\_PASSARGS\_TAIL\_6

```
#define TINYFORMAT_PASSARGS_TAIL_6 , v2, v3, v4, v5, v6
```

Definition at line 468 of file tinyformat.h.

### 7.367.1.58 TINYFORMAT\_PASSARGS\_TAIL\_7

```
#define TINYFORMAT_PASSARGS_TAIL_7 , v2, v3, v4, v5, v6, v7
```

Definition at line 469 of file tinyformat.h.

### 7.367.1.59 TINYFORMAT\_PASSARGS\_TAIL\_8

```
#define TINYFORMAT_PASSARGS_TAIL_8 , v2, v3, v4, v5, v6, v7, v8
```

Definition at line 470 of file tinyformat.h.

### 7.367.1.60 TINYFORMAT\_PASSARGS\_TAIL\_9

```
#define TINYFORMAT_PASSARGS_TAIL_9 , v2, v3, v4, v5, v6, v7, v8, v9
```

Definition at line 471 of file tinyformat.h.

### 7.367.1.61 TINYFORMAT\_VARARGS

```
#define TINYFORMAT_VARARGS(  
    n ) TINYFORMAT_VARARGS_ ## n
```

Definition at line 372 of file tinyformat.h.

### 7.367.1.62 TINYFORMAT\_VARARGS\_1

```
#define TINYFORMAT_VARARGS_1 const T1& v1
```

Definition at line 429 of file tinyformat.h.

### 7.367.1.63 TINYFORMAT\_VARARGS\_10

```
#define TINYFORMAT_VARARGS_10 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5&  
v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10
```

Definition at line 438 of file tinyformat.h.

#### 7.367.1.64 TINYFORMAT\_VARARGS\_11

```
#define TINYFORMAT_VARARGS_11 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11
```

Definition at line 439 of file tinyformat.h.

#### 7.367.1.65 TINYFORMAT\_VARARGS\_12

```
#define TINYFORMAT_VARARGS_12 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12
```

Definition at line 440 of file tinyformat.h.

#### 7.367.1.66 TINYFORMAT\_VARARGS\_13

```
#define TINYFORMAT_VARARGS_13 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12, const T13& v13
```

Definition at line 441 of file tinyformat.h.

#### 7.367.1.67 TINYFORMAT\_VARARGS\_14

```
#define TINYFORMAT_VARARGS_14 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12, const T13& v13, const T14& v14
```

Definition at line 442 of file tinyformat.h.

#### 7.367.1.68 TINYFORMAT\_VARARGS\_15

```
#define TINYFORMAT_VARARGS_15 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12, const T13& v13, const T14& v14, const T15& v15
```

Definition at line 443 of file tinyformat.h.

### 7.367.1.69 TINYFORMAT\_VARARGS\_16

```
#define TINYFORMAT_VARARGS_16 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5, const T6& v6, const T7& v7, const T8& v8, const T9& v9, const T10& v10, const T11& v11, const T12& v12, const T13& v13, const T14& v14, const T15& v15, const T16& v16
```

Definition at line 444 of file tinyformat.h.

### 7.367.1.70 TINYFORMAT\_VARARGS\_2

```
#define TINYFORMAT_VARARGS_2 const T1& v1, const T2& v2
```

Definition at line 430 of file tinyformat.h.

### 7.367.1.71 TINYFORMAT\_VARARGS\_3

```
#define TINYFORMAT_VARARGS_3 const T1& v1, const T2& v2, const T3& v3
```

Definition at line 431 of file tinyformat.h.

### 7.367.1.72 TINYFORMAT\_VARARGS\_4

```
#define TINYFORMAT_VARARGS_4 const T1& v1, const T2& v2, const T3& v3, const T4& v4
```

Definition at line 432 of file tinyformat.h.

### 7.367.1.73 TINYFORMAT\_VARARGS\_5

```
#define TINYFORMAT_VARARGS_5 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5
```

Definition at line 433 of file tinyformat.h.

**7.367.1.74 TINYFORMAT\_VARARGS\_6**

```
#define TINYFORMAT_VARARGS_6 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5,  
const T6& v6
```

Definition at line 434 of file tinyformat.h.

**7.367.1.75 TINYFORMAT\_VARARGS\_7**

```
#define TINYFORMAT_VARARGS_7 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5,  
const T6& v6, const T7& v7
```

Definition at line 435 of file tinyformat.h.

**7.367.1.76 TINYFORMAT\_VARARGS\_8**

```
#define TINYFORMAT_VARARGS_8 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5,  
const T6& v6, const T7& v7, const T8& v8
```

Definition at line 436 of file tinyformat.h.

**7.367.1.77 TINYFORMAT\_VARARGS\_9**

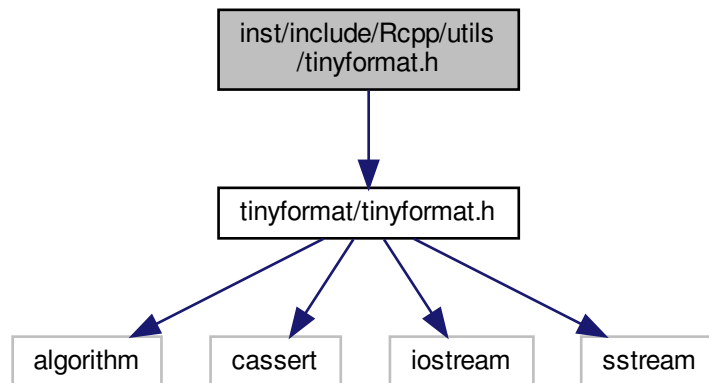
```
#define TINYFORMAT_VARARGS_9 const T1& v1, const T2& v2, const T3& v3, const T4& v4, const T5& v5,  
const T6& v6, const T7& v7, const T8& v8, const T9& v9
```

Definition at line 437 of file tinyformat.h.

## 7.368 inst/include/Rcpp/utis/tinyformat.h File Reference

```
#include "tinyformat/tinyformat.h"
```

Include dependency graph for tinyformat.h:



This graph shows which files directly or indirectly include this file:



### Namespaces

- [Rcpp](#)  
*Rcpp API.*

### Macros

- #define [TINYFORMAT\\_ERROR](#)(REASON) ::[Rcpp::stop](#)(REASON)
- #define [TINYFORMAT\\_NO\\_VARIADIC\\_TEMPLATES](#)
- #define [TINYFORMAT\\_ASSERT](#)(cond) do if (!(cond)) ::[Rcpp::stop](#)("Assertion failed"); while(0)

### Functions

- void [NORET Rcpp::stop](#) (const std::string &message)

## 7.368.1 Macro Definition Documentation

### 7.368.1.1 TINYFORMAT\_ASSERT

```
#define TINYFORMAT_ASSERT(  
    cond ) do if (!(cond)) ::Rcpp::stop("Assertion failed"); while(0)
```

Definition at line 38 of file tinyformat.h.

### 7.368.1.2 TINYFORMAT\_ERROR

```
#define TINYFORMAT_ERROR(  
    REASON ) ::Rcpp::stop(REASON)
```

Definition at line 29 of file tinyformat.h.

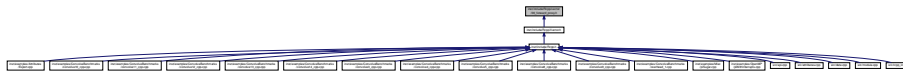
### 7.368.1.3 TINYFORMAT\_NO\_VARIADIC\_TEMPLATES

```
#define TINYFORMAT_NO_VARIADIC_TEMPLATES
```

Definition at line 35 of file tinyformat.h.

## 7.369 inst/include/Rcpp/vector/00\_forward\_proxy.h File Reference

This graph shows which files directly or indirectly include this file:



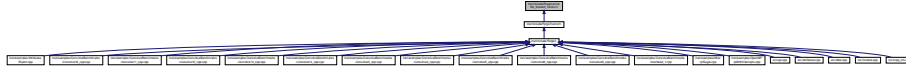
## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*



## 7.370 inst/include/Rcpp/vector/00\_forward\_Vector.h File Reference

This graph shows which files directly or indirectly include this file:

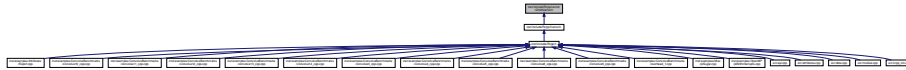


### Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.371 inst/include/Rcpp/vector/ChildVector.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

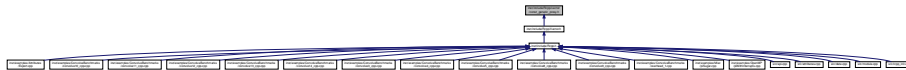
- class [Rcpp::ChildVector< T >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.372 inst/include/Rcpp/vector/const\_generic\_proxy.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

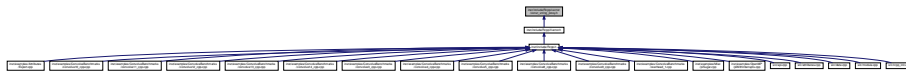
- class [Rcpp::internal::const\\_generic\\_proxy< RTYPE, StoragePolicy >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## 7.373 inst/include/Rcpp/vector/const\_string\_proxy.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::internal::const\\_string\\_proxy< RTYPE, StoragePolicy >](#)

## Namespaces

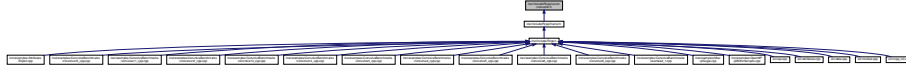
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Functions

- `template<int RT>`  
`bool Rcpp::internal::operator< (const const_string_proxy< RT > &lhs, const const_string_proxy< RT > &rhs)`
- `template<int RT>`  
`bool Rcpp::internal::operator> (const const_string_proxy< RT > &lhs, const const_string_proxy< RT > &rhs)`
- `template<int RT>`  
`bool Rcpp::internal::operator>= (const const_string_proxy< RT > &lhs, const const_string_proxy< RT > &rhs)`
- `template<int RT>`  
`bool Rcpp::internal::operator<= (const const_string_proxy< RT > &lhs, const const_string_proxy< RT > &rhs)`
- `template<template< class > class StoragePolicy>`  
`std::ostream & Rcpp::internal::operator<< (std::ostream &os, const const_string_proxy< STRSXP, StoragePolicy > &proxy)`
- `template<template< class > class StoragePolicy>`  
`std::string Rcpp::internal::operator+ (const std::string &x, const const_string_proxy< STRSXP, StoragePolicy > &y)`

## 7.374 inst/include/Rcpp/vector/converter.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

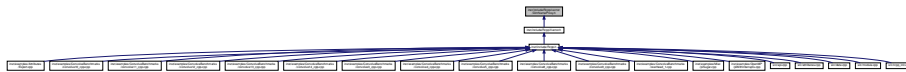
- class [Rcpp::internal::element\\_converter< RTYPE >](#)
- class [Rcpp::internal::string\\_element\\_converter< RTYPE >](#)
- class [Rcpp::internal::generic\\_element\\_converter< RTYPE >](#)
- struct [Rcpp::traits::r\\_vector\\_element\\_converter< RTYPE >](#)
- struct [Rcpp::traits::r\\_vector\\_element\\_converter< STRSXP >](#)
- struct [Rcpp::traits::r\\_vector\\_element\\_converter< VECSXP >](#)
- struct [Rcpp::traits::r\\_vector\\_element\\_converter< EXPRSXP >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*

## 7.375 inst/include/Rcpp/vector/DimNameProxy.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

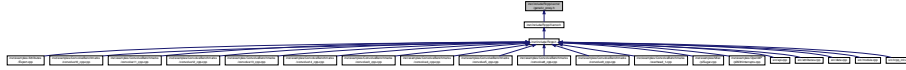
- class [Rcpp::internal::DimNameProxy](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## 7.376 inst/include/Rcpp/vector/generic\_proxy.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

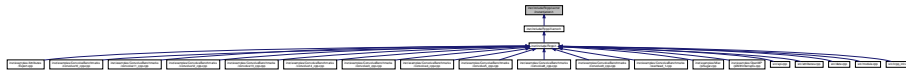
- class [Rcpp::internal::generic\\_proxy< RTYPE, StoragePolicy >](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## 7.377 inst/include/Rcpp/vector/instantiation.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

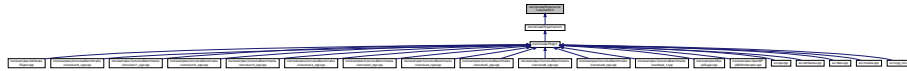
- [Rcpp](#)  
*Rcpp API.*

## Typedefs

- typedef Vector< CPLXSXP > [Rcpp::ComplexVector](#)
- typedef Vector< INTSXP > [Rcpp::IntegerVector](#)
- typedef Vector< LGLSXP > [Rcpp::LogicalVector](#)
- typedef Vector< REALSXP > [Rcpp::NumericVector](#)
- typedef Vector< REALSXP > [Rcpp::DoubleVector](#)
- typedef Vector< RAWSXP > [Rcpp::RawVector](#)
- typedef Vector< STRSXP > [Rcpp::CharacterVector](#)
- typedef Vector< STRSXP > [Rcpp::StringVector](#)
- typedef Vector< VECSXP > [Rcpp::GenericVector](#)
- typedef Vector< VECSXP > [Rcpp::List](#)
- typedef Vector< EXPRSXP > [Rcpp::ExpressionVector](#)
- typedef Matrix< CPLXSXP > [Rcpp::ComplexMatrix](#)
- typedef Matrix< INTSXP > [Rcpp::IntegerMatrix](#)
- typedef Matrix< LGLSXP > [Rcpp::LogicalMatrix](#)
- typedef Matrix< REALSXP > [Rcpp::NumericMatrix](#)
- typedef Matrix< RAWSXP > [Rcpp::RawMatrix](#)
- typedef Matrix< STRSXP > [Rcpp::CharacterMatrix](#)
- typedef Matrix< STRSXP > [Rcpp::StringMatrix](#)
- typedef Matrix< VECSXP > [Rcpp::GenericMatrix](#)
- typedef Matrix< VECSXP > [Rcpp::ListMatrix](#)
- typedef Matrix< EXPRSXP > [Rcpp::ExpressionMatrix](#)

## 7.378 inst/include/Rcpp/vector/LazyVector.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

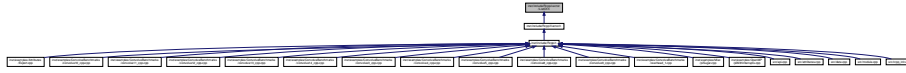
- class [Rcpp::internal::LazyVector< VECTOR >](#)
- class [Rcpp::internal::LazyVector< Rcpp::Vector< RTYPE > >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## 7.379 inst/include/Rcpp/vector/ListOf.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::ListOf< T >](#)

### Namespaces

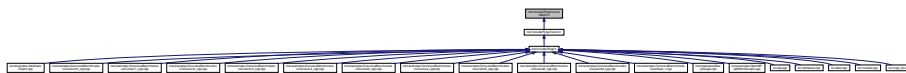
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::sugar](#)

### Functions

- [template<typename T , typename Function >](#)  
List [Rcpp::lapply](#) (const ListOf< T > &t, Function fun)
- [template<typename T , typename Function >](#)  
T [Rcpp::sapply](#) (const ListOf< T > &t, Function fun)

## 7.380 inst/include/Rcpp/vector/Matrix.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::Matrix< RTYPE, StoragePolicy >](#)
- struct [Rcpp::Matrix< RTYPE, StoragePolicy >::r\\_type](#)
- struct [Rcpp::Matrix< RTYPE, StoragePolicy >::can\\_have\\_na](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Macros

- #define [RCPP\\_GENERATE\\_MATRIX\\_SCALAR\\_OPERATOR](#)(\_\_OPERATOR\_\_)
- #define [RCPP\\_GENERATE\\_SCALAR\\_MATRIX\\_OPERATOR](#)(\_\_OPERATOR\_\_)

## Functions

- internal::DimNameProxy [Rcpp::rownames](#) (SEXP x)
- internal::DimNameProxy [Rcpp::colnames](#) (SEXP x)
- template<template< class > class StoragePolicy>  
std::ostream & [Rcpp::operator<<](#) (std::ostream &s, const Matrix< REALSXP, StoragePolicy > &rhs)
- template<template< class > class StoragePolicy>  
std::ostream & [Rcpp::operator<<](#) (std::ostream &s, const Matrix< INTSXP, StoragePolicy > &rhs)
- template<template< class > class StoragePolicy>  
std::ostream & [Rcpp::operator<<](#) (std::ostream &s, const Matrix< STRSXP, StoragePolicy > &rhs)
- template<int RTYPE, template< class > class StoragePolicy>  
std::ostream & [Rcpp::operator<<](#) (std::ostream &s, const Matrix< RTYPE, StoragePolicy > &rhs)
- template<int RTYPE, template< class > class StoragePolicy>  
Matrix< RTYPE, StoragePolicy > [Rcpp::transpose\\_impl](#) (const Matrix< RTYPE, StoragePolicy > &x)
- template<template< class > class StoragePolicy>  
Matrix< REALSXP, StoragePolicy > [Rcpp::transpose](#) (const Matrix< REALSXP, StoragePolicy > &x)
- template<template< class > class StoragePolicy>  
Matrix< INTSXP, StoragePolicy > [Rcpp::transpose](#) (const Matrix< INTSXP, StoragePolicy > &x)
- template<template< class > class StoragePolicy>  
Matrix< STRSXP, StoragePolicy > [Rcpp::transpose](#) (const Matrix< STRSXP, StoragePolicy > &x)

## 7.380.1 Macro Definition Documentation

### 7.380.1.1 RCPP\_GENERATE\_MATRIX\_SCALAR\_OPERATOR

```
#define RCPP_GENERATE_MATRIX_SCALAR_OPERATOR(  
    __OPERATOR__ )
```

#### Value:

```
template <int RTYPE, template <class> class StoragePolicy, typename T >  
inline typename traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< T  
>::type,  
    typename Matrix<RTYPE, StoragePolicy>::stored_type >::value, Matrix<RTYPE, StoragePolicy> >::type  
    operator __OPERATOR__ (const Matrix<RTYPE, StoragePolicy> &lhs, const T &rhs) {  
    Vector<RTYPE, StoragePolicy> v = static_cast<const Vector<RTYPE, StoragePolicy> &>(lhs) __OPERATOR__  
    rhs;  
    v.attr("dim") = Vector<INTSXP>::create(lhs.nrow(), lhs.ncol());  
    return as< Matrix<RTYPE, StoragePolicy> >(v);  
}
```

Definition at line 250 of file Matrix.h.

### 7.380.1.2 RCPP\_GENERATE\_SCALAR\_MATRIX\_OPERATOR

```
#define RCPP_GENERATE_SCALAR_MATRIX_OPERATOR(  
    __OPERATOR__ )
```

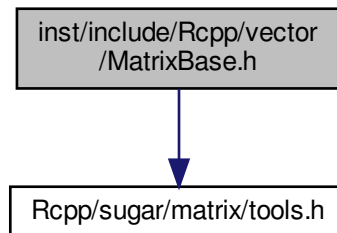
#### Value:

```
template <int RTYPE, template <class> class StoragePolicy, typename T >  
inline typename traits::enable_if< traits::is_convertible< typename traits::remove_const_and_reference< T  
>::type,  
    typename Matrix<RTYPE, StoragePolicy>::stored_type >::value, Matrix<RTYPE, StoragePolicy> >::type  
    operator __OPERATOR__ (const T &lhs, const Matrix<RTYPE, StoragePolicy> &rhs) {  
    Vector<RTYPE, StoragePolicy> v = lhs __OPERATOR__ static_cast<const Vector<RTYPE, StoragePolicy>  
&>(rhs);  
    v.attr("dim") = Vector<INTSXP>::create(rhs.nrow(), rhs.ncol());  
    return as< Matrix<RTYPE, StoragePolicy> >(v);  
}
```

Definition at line 267 of file Matrix.h.

## 7.381 inst/include/Rcpp/vector/MatrixBase.h File Reference

```
#include <Rcpp/sugar/matrix/tools.h>  
Include dependency graph for MatrixBase.h:
```



This graph shows which files directly or indirectly include this file:





## Classes

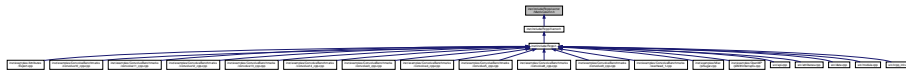
- class [Rcpp::MatrixBase< RTYPE, na, MATRIX >](#)
- struct [Rcpp::MatrixBase< RTYPE, na, MATRIX >::r\\_type](#)
- struct [Rcpp::MatrixBase< RTYPE, na, MATRIX >::r\\_matrix\\_interface](#)
- struct [Rcpp::MatrixBase< RTYPE, na, MATRIX >::can\\_have\\_na](#)
- class [Rcpp::MatrixBase< RTYPE, na, MATRIX >::iterator](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.382 inst/include/Rcpp/vector/MatrixColumn.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

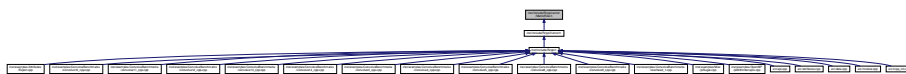
- class [Rcpp::MatrixColumn< RTYPE >](#)
- class [Rcpp::ConstMatrixColumn< RTYPE >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.383 inst/include/Rcpp/vector/MatrixRow.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

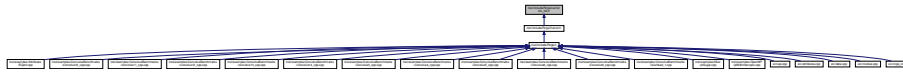
- class [Rcpp::MatrixRow< RTYPE >](#)
- struct [Rcpp::MatrixRow< RTYPE >::iter\\_traits](#)
- struct [Rcpp::MatrixRow< RTYPE >::const\\_iter\\_traits](#)
- class [Rcpp::MatrixRow< RTYPE >::iter\\_base< TRAITS >](#)
- class [Rcpp::ConstMatrixRow< RTYPE >](#)
- class [Rcpp::ConstMatrixRow< RTYPE >::const\\_iterator](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.384 inst/include/Rcpp/vector/no\_init.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::no\\_init\\_vector](#)
- class [Rcpp::no\\_init\\_matrix](#)

## Namespaces

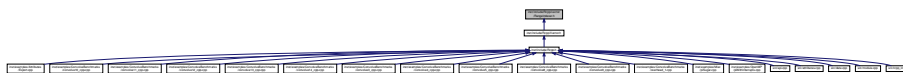
- [Rcpp](#)  
*Rcpp API.*

## Functions

- `no_init_vector` [Rcpp::no\\_init](#) (R\_xlen\_t size)
- `no_init_matrix` [Rcpp::no\\_init](#) (int nr, int nc)

## 7.385 inst/include/Rcpp/vector/RangeIndexer.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::internal::RangeIndexer< RTYPE, NA, VECTOR >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Macros

- #define [UNROLL\\_LOOP\(OP\)](#)

### 7.385.1 Macro Definition Documentation

#### 7.385.1.1 UNROLL\_LOOP

```
#define UNROLL_LOOP(  
    OP )
```

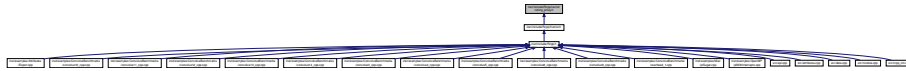
#### Value:

```
typedef typename ::Rcpp::traits::Extractor<RTYPE,NA,T>::type EXT ; \
const EXT& input( x.get_ref() ) ;  
int __trip_count = (size_) » 2;  
int i=0 ;  
for ( ; __trip_count > 0 ; --__trip_count) {  
    start[i] OP input[i] ; i++ ;  
    start[i] OP input[i] ; i++ ;  
    start[i] OP input[i] ; i++ ;  
    start[i] OP input[i] ; i++ ;  
}  
switch (size_ - i){  
    case 3:  
        start[i] OP input[i] ; i++ ;  
    case 2:  
        start[i] OP input[i] ; i++ ;  
    case 1:  
        start[i] OP input[i] ; i++ ;  
    case 0:  
    default:  
        return *this ;  
}
```

Definition at line 25 of file RangeIndexer.h.

## 7.386 inst/include/Rcpp/vector/string\_proxy.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::internal::string\\_proxy< RTYPE, StoragePolicy >](#)

### Namespaces

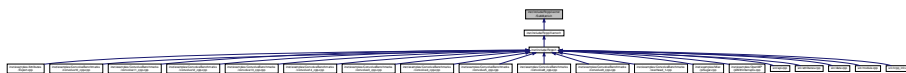
- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

### Functions

- `template<int RT>`  
bool [Rcpp::internal::operator<](#) (const string\_proxy< RT > &lhs, const string\_proxy< RT > &rhs)
- `template<int RT>`  
bool [Rcpp::internal::operator>](#) (const string\_proxy< RT > &lhs, const string\_proxy< RT > &rhs)
- `template<int RT>`  
bool [Rcpp::internal::operator>=](#) (const string\_proxy< RT > &lhs, const string\_proxy< RT > &rhs)
- `template<int RT>`  
bool [Rcpp::internal::operator<=](#) (const string\_proxy< RT > &lhs, const string\_proxy< RT > &rhs)
- `std::ostream &` [Rcpp::internal::operator<<](#) (std::ostream &os, const string\_proxy< STRSXP > &proxy)
- `std::string` [Rcpp::internal::operator+](#) (const std::string &x, const string\_proxy< STRSXP > &y)

## 7.387 inst/include/Rcpp/vector/SubMatrix.h File Reference

This graph shows which files directly or indirectly include this file:



### Classes

- class [Rcpp::SubMatrix< RTYPE >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Macros

- #define [RCPP\\_WRAP\\_SUBMATRIX](#)(RTYPE)

### 7.387.1 Macro Definition Documentation

#### 7.387.1.1 RCPP\_WRAP\_SUBMATRIX

```
#define RCPP_WRAP_SUBMATRIX(  
    RTYPE )
```

#### Value:

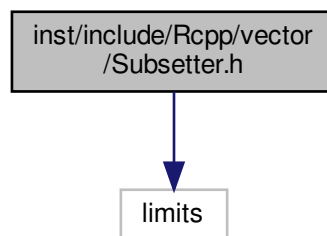
```
template<> inline SEXP wrap< SubMatrix<RTYPE> >( \
    const SubMatrix<RTYPE>& object           \
) {                                         \
    return Matrix<RTYPE>( object ) ;      \
}
```

Definition at line 90 of file SubMatrix.h.

## 7.388 inst/include/Rcpp/vector/Subsetter.h File Reference

```
#include <limits>
```

Include dependency graph for Subsetter.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::SubsetProxy< RTYPE, StoragePolicy, RHS\\_RTYPE, RHS\\_NA, RHS\\_T >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Macros

- #define [RCPP\\_GENERATE\\_SUBSET\\_PROXY\\_OPERATOR\(\\_\\_OPERATOR\\_\\_\)](#)

### 7.388.1 Macro Definition Documentation

#### 7.388.1.1 RCPP\_GENERATE\_SUBSET\_PROXY\_OPERATOR

```
#define RCPP_GENERATE_SUBSET_PROXY_OPERATOR(  
    __OPERATOR__ )
```

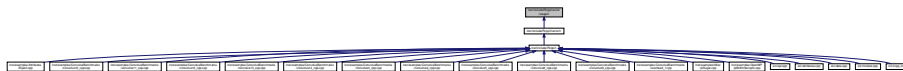
#### Value:

```
template <int RTYPE_OTHER, template <class> class StoragePolicyOther,  
    int RHS_RTYPE_OTHER, bool RHS_NA_OTHER, typename RHS_T_OTHER>  
Vector<RTYPE, StoragePolicy> operator __OPERATOR__ (  
    const SubsetProxy<RTYPE_OTHER, StoragePolicyOther, RHS_RTYPE_OTHER,  
        RHS_NA_OTHER, RHS_T_OTHER>& other) {  
    Vector<RTYPE, StoragePolicy> result(indices_n);  
    if (other.indices_n == 1) {  
        for (R_xlen_t i = 0; i < indices_n; ++i)  
            result[i] = lhs[indices[i]] __OPERATOR__ other.lhs[other.indices[0]];  
    } else if (indices_n == other.indices_n) {  
        for (R_xlen_t i = 0; i < indices_n; ++i)  
            result[i] = lhs[indices[i]] __OPERATOR__ other.lhs[other.indices[i]];  
    } else {  
        stop("index error");  
    }  
    return result;  
}
```

Definition at line 237 of file Subsetter.h.

### 7.389 inst/include/Rcpp/vector/swap.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [std](#)

## Macros

- #define [RCPP\\_GENERATE\\_SWAP](#)(TYPE, RTYPE)

### 7.389.1 Macro Definition Documentation

#### 7.389.1.1 RCPP\_GENERATE\_SWAP

```
#define RCPP_GENERATE_SWAP (
    TYPE,
    RTYPE )
```

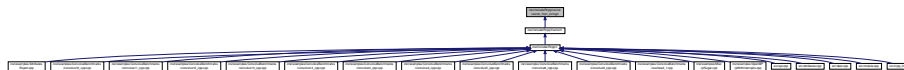
#### Value:

```
template<> inline void swap< Rcpp::internal::TYPE<RTYPE> >( \
    Rcpp::internal::TYPE<RTYPE>& a , \
    Rcpp::internal::TYPE<RTYPE>& b) { \
    a.swap(b) ; \
}
```

Definition at line 28 of file swap.h.

## 7.390 inst/include/Rcpp/vector/vector\_from\_string.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Functions

- `template<int RTYPE>`  
SEXP [Rcpp::internal::vector\\_from\\_string](#) (const std::string &st)
- `template<int RTYPE>`  
SEXP [Rcpp::internal::vector\\_from\\_string\\_expr](#) (const std::string &code)
- `template<>` SEXP [Rcpp::internal::vector\\_from\\_string< EXPRXP >](#) (const std::string &st)

## 7.391 inst/include/Rcpp/vector/VectorBase.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

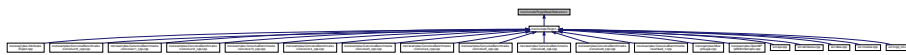
- class [Rcpp::VectorBase< RTYPE, na, VECTOR >](#)
- struct [Rcpp::VectorBase< RTYPE, na, VECTOR >::rcpp\\_sugar\\_expression](#)
- struct [Rcpp::VectorBase< RTYPE, na, VECTOR >::r\\_type](#)
- struct [Rcpp::VectorBase< RTYPE, na, VECTOR >::can\\_have\\_na](#)
- struct [Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\\_traits](#)
- struct [Rcpp::VectorBase< RTYPE, na, VECTOR >::const\\_iter\\_traits](#)
- class [Rcpp::VectorBase< RTYPE, na, VECTOR >::iter\\_base< TRAITS >](#)

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## 7.392 inst/include/Rcpp/WeakReference.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*



## Typedefs

- typedef WeakReference\_Impl< PreserveStorage > [Rcpp::WeakReference](#)

## Functions

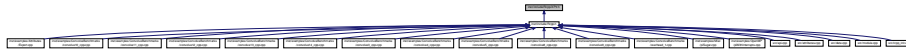
- [Rcpp::RCPP\\_API\\_CLASS](#) (WeakReference\_Impl)

## 7.393 inst/include/Rcpp/XPtr.h File Reference

```
#include <RcppCommon.h>
Include dependency graph for XPtr.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [Rcpp::XPtr](#)< T, StoragePolicy, Finalizer, finalizeOnExit >

## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Functions

- template<typename T >  
void [Rcpp::standard\\_delete\\_finalizer](#) (T \*obj)
- template<typename T, void Finalizer>  
void [Rcpp::finalizer\\_wrapper](#) (SEXP p)

## 7.394 inst/include/RcppCommon.h File Reference

```
#include <Rcpp/r/headers.h>
#include <iterator>
#include <exception>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <string>
#include <list>
#include <map>
#include <set>
#include <stdexcept>
#include <vector>
#include <deque>
#include <functional>
#include <numeric>
#include <algorithm>
#include <complex>
#include <cfloat>
#include <limits>
#include <typeinfo>
#include <utility>
#include <Rcpp/sprintf.h>
#include <R_ext/Callbacks.h>
#include <R_ext/Visibility.h>
#include <Rcpp/utils/tinyformat.h>
#include <Rmath.h>
#include <Rcpp/sugar/undoRmath.h>
#include <Rcpp/storage/storage.h>
#include <Rcpp/protection/protection.h>
#include <Rcpp/routines.h>
#include <Rcpp/exceptions.h>
#include <Rcpp/proxy/proxy.h>
#include <Rcpp/lang.h>
#include <Rcpp/complex.h>
#include <Rcpp/barrier.h>
#include <Rcpp/Interrupt.h>
#include <Rcpp/longlong.h>
#include <Rcpp/internal/na.h>
#include <Rcpp/internal/NAComparator.h>
#include <Rcpp/internal/NAEquals.h>
#include <Rcpp/traits/traits.h>
#include <Rcpp/Named.h>
#include <Rcpp/internal/caster.h>
#include <Rcpp/internal/r_vector.h>
#include <Rcpp/r_cast.h>
#include <Rcpp/api/bones/bones.h>
#include <Rcpp/internal/export.h>
#include <Rcpp/internal/r_coerce.h>
#include <Rcpp/as.h>
#include <Rcpp/InputParameter.h>
#include <Rcpp/is.h>
#include <Rcpp/vector/VectorBase.h>
```

```

#include <Rcpp/vector/MatrixBase.h>
#include <Rcpp/internal/ListInitialization.h>
#include <Rcpp/internal/Proxy_Iterator.h>
#include <Rcpp/internal/SEXP_Iterator.h>
#include <Rcpp/internal/converter.h>
#include <Rcpp/print.h>
#include <Rcpp/algo.h>
#include <Rcpp/sugar/sugar_forward.h>
#include <Rcpp/iostream/Rstreambuf.h>
#include <Rcpp/internal/wrap.h>

```

This graph shows which files directly or indirectly include this file:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::traits](#)  
*traits used to dispatch wrap*
- [Rcpp::internal](#)  
*internal implementation details*

## Macros

- `#define RcppExport extern "C" attribute_visible`

## Functions

- SEXP [Rcpp::Rcpp\\_fast\\_eval](#) (SEXP expr, SEXP env)
- SEXP [Rcpp::Rcpp\\_eval](#) (SEXP expr, SEXP env)
- attribute\_hidden SEXP [Rcpp::Rcpp\\_precious\\_preserve](#) (SEXP object)
- attribute\_hidden void [Rcpp::Rcpp\\_precious\\_remove](#) (SEXP token)
- SEXP [Rcpp::internal::Rcpp\\_eval\\_impl](#) (SEXP expr, SEXP env)
- SEXP [Rcpp::Rcpp\\_PreserveObject](#) (SEXP x)
- void [Rcpp::Rcpp\\_ReleaseObject](#) (SEXP x)
- SEXP [Rcpp::Rcpp\\_ReplaceObject](#) (SEXP x, SEXP y)
- SEXP [Rcpp::Rcpp\\_PreciousPreserve](#) (SEXP object)
- void [Rcpp::Rcpp\\_PreciousRelease](#) (SEXP token)
- template<typename Class >  
SEXP [Rcpp::internal::make\\_new\\_object](#) (Class \*ptr)

### 7.394.1 Macro Definition Documentation

### 7.394.1.1 RcppExport

```
#define RcppExport extern "C" attribute_visible
```

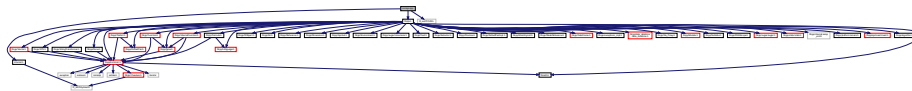
#### Examples

[ConvolveBenchmarks/convolve3\\_cpp.cpp](#), and [ConvolveBenchmarks/convolve4\\_cpp.cpp](#).

Definition at line 140 of file RcppCommon.h.

## 7.395 src/api.cpp File Reference

```
#include <Rcpp.h>
#include "internal.h"
#include <R_ext/PrtUtil.h>
Include dependency graph for api.cpp:
```



### Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

### Macros

- #define [COMPILING\\_RCPP](#)
- #define [RCPP\\_USE\\_GLOBAL\\_ROSTREAM](#)

### Functions

- `attribute_hidden Rostream< true > & Rcpp::Rcpp_cout_get ()`
- `attribute_hidden Rostream< false > & Rcpp::Rcpp_cerr_get ()`
- `attribute_hidden unsigned long Rcpp::internal::enterRNGScope ()`
- `attribute_hidden unsigned long Rcpp::internal::exitRNGScope ()`
- `attribute_hidden unsigned long Rcpp::internal::beginSuspendRNGSynchronization ()`
- `attribute_hidden unsigned long Rcpp::internal::endSuspendRNGSynchronization ()`
- `attribute_hidden char * Rcpp::internal::get_string_buffer ()`
- `attribute_hidden const char * Rcpp::type2name (SEXP x)`
- `std::string demangle (const std::string &name)`
- `const char * short_file_name (const char *file)`
- `SEXP as_character_externalptr (SEXP xp)`
- `SEXP rcpp_capabilities ()`
- `SEXP rcpp_can_use_cxx0x ()`
- `SEXP rcpp_can_use_cxx11 ()`
- `SEXP stack_trace (const char *file, int line)`
- `SEXP getRcppVersionStrings ()`

## Variables

- int `Rcpp::internal::rngSynchronizationSuspended` = 0

## 7.395.1 Macro Definition Documentation

### 7.395.1.1 COMPILING\_RCPP

```
#define COMPILING_RCPP
```

Definition at line 23 of file api.cpp.

### 7.395.1.2 RCPP\_USE\_GLOBAL\_ROSTREAM

```
#define RCPP_USE_GLOBAL_ROSTREAM
```

Definition at line 24 of file api.cpp.

## 7.395.2 Function Documentation

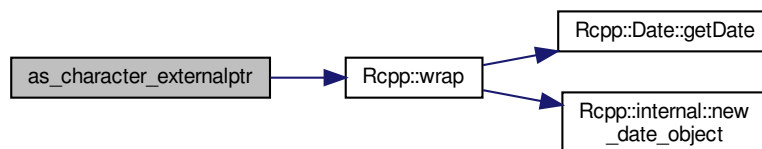
### 7.395.2.1 as\_character\_externalptr()

```
SEXP as_character_externalptr (  
    SEXP xp )
```

Definition at line 160 of file api.cpp.

References `Rcpp::wrap()`.

Here is the call graph for this function:



### 7.395.2.2 demangle()

```
std::string demangle (
    const std::string & name ) [inline]
```

Definition at line 125 of file api.cpp.

References GET\_CALLABLE.

Referenced by exception\_to\_condition\_template(), Rcpp::get\_return\_type\_dispatch(), and registerFunctions().

### 7.395.2.3 getRcppVersionStrings()

```
SEXP getRcppVersionStrings ( )
```

Definition at line 289 of file api.cpp.

References RCPP\_DEV\_VERSION\_STRING, and RCPP\_VERSION\_STRING.

### 7.395.2.4 rcpp\_can\_use\_cxx0x()

```
SEXP rcpp_can_use_cxx0x ( )
```

Definition at line 255 of file api.cpp.

### 7.395.2.5 rcpp\_can\_use\_cxx11()

```
SEXP rcpp_can_use_cxx11 ( )
```

Definition at line 265 of file api.cpp.

### 7.395.2.6 rcpp\_capabilities()

```
SEXP rcpp_capabilities ( )
```

Definition at line 167 of file api.cpp.

### 7.395.2.7 short\_file\_name()

```
const char* short_file_name (
    const char * file )
```

Definition at line 149 of file api.cpp.

### 7.395.2.8 stack\_trace()

```
SEXP stack_trace (
    const char * file,
    int line )
```

Definition at line 275 of file api.cpp.

## 7.396 src/attributes.cpp File Reference

```
#include <sys/types.h>
#include <sys/stat.h>
#include <errno.h>
#include <cstring>
#include <string>
#include <vector>
#include <map>
#include <set>
#include <algorithm>
#include <fstream>
#include <sstream>
#include <Rcpp.h>
```

Include dependency graph for attributes.cpp:



## Classes

- class [Rcpp::attributes::FileInfo](#)
- class [Rcpp::attributes::Type](#)
- class [Rcpp::attributes::Argument](#)
- class [Rcpp::attributes::Function](#)
- class [Rcpp::attributes::Param](#)
- class [Rcpp::attributes::Attribute](#)
- class [Rcpp::attributes::SourceFileAttributes](#)
- class [Rcpp::attributes::CommentState](#)

- class [Rcpp::attributes::SourceFileAttributesParser](#)
- class [Rcpp::attributes::ExportsGenerator](#)
- class [Rcpp::attributes::CppExportsGenerator](#)
- class [Rcpp::attributes::CppExportsIncludeGenerator](#)
- class [Rcpp::attributes::CppPackageIncludeGenerator](#)
- class [Rcpp::attributes::REExportsGenerator](#)
- class [Rcpp::attributes::ExportsGenerators](#)

## Namespaces

- [Rcpp](#)
  - Rcpp API.*
- [Rcpp::attributes](#)

## Macros

- `#define COMPILING_RCPP`
- `#define RCPP_NO_SUGAR`

## Functions

- bool [Rcpp::attributes::removeFile](#) (const std::string &path)
- void [Rcpp::attributes::createDirectory](#) (const std::string &path)
- bool [Rcpp::attributes::isWhitespace](#) (char ch)
- void [Rcpp::attributes::trimWhitespace](#) (std::string \*pStr)
- void [Rcpp::attributes::stripTrailingLineComments](#) (std::string \*pStr)
- void [Rcpp::attributes::stripQuotes](#) (std::string \*pStr)
- bool [Rcpp::attributes::isQuoted](#) (const std::string &str)
- bool [Rcpp::attributes::endsWith](#) (const std::string &str, const std::string &suffix)
- void [Rcpp::attributes::showWarning](#) (const std::string &msg)
- bool [Rcpp::attributes::isRxygenCpp](#) (const std::string &str)
- std::ostream & [Rcpp::attributes::operator<<](#) (std::ostream &os, const [Type](#) &type)
- std::ostream & [Rcpp::attributes::operator<<](#) (std::ostream &os, const [Argument](#) &argument)
- std::ostream & [Rcpp::attributes::operator<<](#) (std::ostream &os, const [Function](#) &function)
- std::ostream & [Rcpp::attributes::operator<<](#) (std::ostream &os, const [Param](#) &param)
- std::ostream & [Rcpp::attributes::operator<<](#) (std::ostream &os, const [Attribute](#) &attribute)
- std::string [Rcpp::attributes::generateRArgList](#) (const [Function](#) &function)
- bool [Rcpp::attributes::checkRSignature](#) (const [Function](#) &function, std::string args)
- void [Rcpp::attributes::initializeGlobals](#) (std::ostream &ostr)
- void [Rcpp::attributes::generateCpp](#) (std::ostream &ostr, const [SourceFileAttributes](#) &attributes, bool include← Prototype, bool cppInterface, const std::string &contextId)
- void [Rcpp::attributes::printArgument](#) (std::ostream &os, const [Argument](#) &argument, bool printDefault=true)
- void [Rcpp::attributes::printFunction](#) (std::ostream &os, const [Function](#) &function, bool printArgDefaults=true)
- [RcppExport](#) SEXP [sourceCppContext](#) (SEXP sFile, SEXP sCode, SEXP sRebuild, SEXP sCacheDir, SEXP s← Platform)
- [RcppExport](#) SEXP [compileAttributes](#) (SEXP sPackageDir, SEXP sPackageName, SEXP sDepends, SEXP s← Registration, SEXP sCppFiles, SEXP sCppFileBasenames, SEXP sIncludes, SEXP sVerbose, SEXP sPlatform)



## Variables

- const char \*const [Rcpp::attributes::kWhitespaceChars](#) = "\n\r\t\v"
- const char \*const [Rcpp::attributes::kExportAttribute](#) = "export"
- const char \*const [Rcpp::attributes::kExportName](#) = "name"
- const char \*const [Rcpp::attributes::kExportRng](#) = "rng"
- const char \*const [Rcpp::attributes::kExportInvisible](#) = "invisible"
- const char \*const [Rcpp::attributes::kExportSignature](#) = "signature"
- const char \*const [Rcpp::attributes::kInitAttribute](#) = "init"
- const char \*const [Rcpp::attributes::kDependsAttribute](#) = "depends"
- const char \*const [Rcpp::attributes::kPluginsAttribute](#) = "plugins"
- const char \*const [Rcpp::attributes::kInterfacesAttribute](#) = "interfaces"
- const char \*const [Rcpp::attributes::kInterfaceR](#) = "r"
- const char \*const [Rcpp::attributes::kInterfaceCpp](#) = "cpp"
- const char \*const [Rcpp::attributes::kParamValueFalse](#) = "false"
- const char \*const [Rcpp::attributes::kParamValueTrue](#) = "true"
- const char \*const [Rcpp::attributes::kParamValueFALSE](#) = "FALSE"
- const char \*const [Rcpp::attributes::kParamValueTRUE](#) = "TRUE"
- const char \*const [Rcpp::attributes::kParamBlockStart](#) = "{;"
- const char \*const [Rcpp::attributes::kParamBlockEnd](#) = "}"

## 7.396.1 Macro Definition Documentation

### 7.396.1.1 COMPILING\_RCPP

```
#define COMPILING_RCPP
```

Definition at line 22 of file attributes.cpp.

### 7.396.1.2 RCPP\_NO\_SUGAR

```
#define RCPP_NO_SUGAR
```

Definition at line 38 of file attributes.cpp.

## 7.396.2 Function Documentation

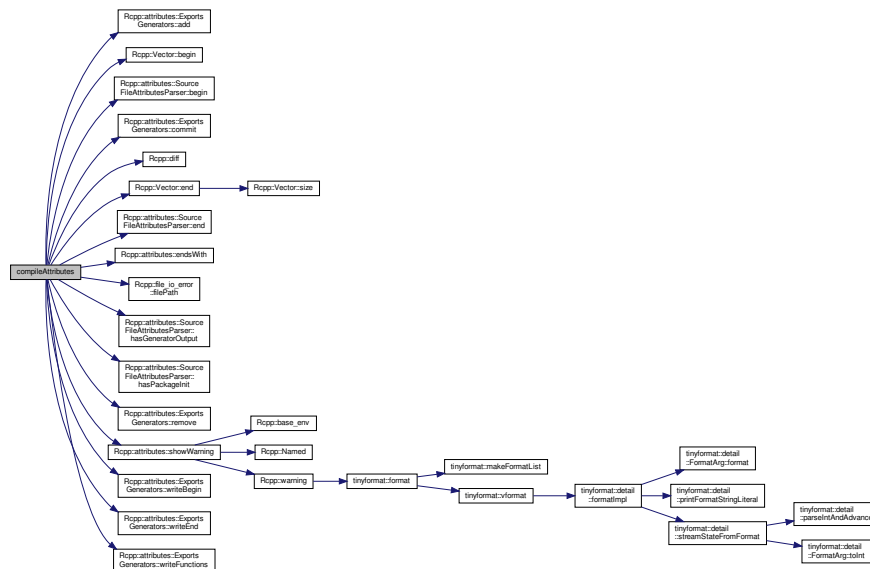
### 7.396.2.1 compileAttributes()

```
RcppExport SEXP compileAttributes (
    SEXP sPackageDir,
    SEXP sPackageName,
    SEXP sDepends,
    SEXP sRegistration,
    SEXP sCppFiles,
    SEXP sCppFileBasenames,
    SEXP sIncludes,
    SEXP sVerbose,
    SEXP sPlatform )
```

Definition at line 3620 of file attributes.cpp.

References `Rcpp::attributes::ExportsGenerators::add()`, `Rcpp::Vector< RTYPE, StoragePolicy >::begin()`, `Rcpp::attributes::SourceFileAttributesParser::begin()`, `BEGIN_RCPP`, `Rcpp::attributes::ExportsGenerators::commit()`, `Rcpp::diff()`, `Rcpp::Vector< RTYPE, StoragePolicy >::end()`, `Rcpp::attributes::SourceFileAttributesParser::end()`, `END_RCPP`, `Rcpp::attributes::endsWith()`, `Rcpp::file_io_error::filePath()`, `Rcpp::attributes::SourceFileAttributesParser::hasGeneratorOutput()`, `Rcpp::attributes::SourceFileAttributesParser::hasPackageInit()`, `Rcpp::attributes::kDependsAttribute`, `Rcpp::Rcout`, `Rcpp::attributes::ExportsGenerators::remove()`, `Rcpp::attributes::showWarning()`, `Rcpp::attributes::ExportsGenerators::writeBegin()`, `Rcpp::attributes::ExportsGenerators::writeEnd()`, and `Rcpp::attributes::ExportsGenerators::writeFunctions()`.

Here is the call graph for this function:



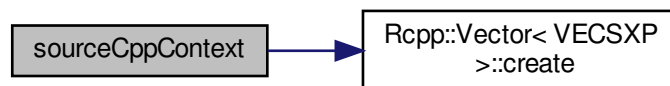
## 7.396.2.2 sourceCppContext()

```
RcppExport SEXP sourceCppContext (
    SEXP sFile,
    SEXP sCode,
    SEXP sRebuild,
    SEXP sCacheDir,
    SEXP sPlatform )
```

Definition at line 3556 of file attributes.cpp.

References Rcpp::\_, BEGIN\_RCPP, Rcpp::Vector< VECSXP >::create(), and END\_RCPP.

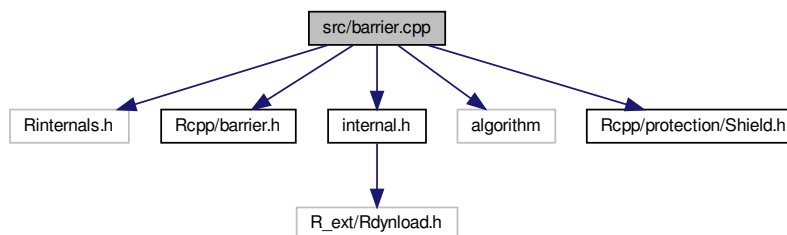
Here is the call graph for this function:



## 7.397 src/barrier.cpp File Reference

```
#include <Rinternals.h>
#include <Rcpp/barrier.h>
#include "internal.h"
#include <algorithm>
#include <Rcpp/protection/Shield.h>
```

Include dependency graph for barrier.cpp:



## Namespaces

- [Rcpp](#)  
*Rcpp API.*
- [Rcpp::internal](#)  
*internal implementation details*

## Macros

- #define [COMPILING\\_RCPP](#)
- #define [USE\\_RINTERNALS](#)
- #define [RCPP\\_HASH\\_CACHE\\_INDEX](#) 4
- #define [RCPP\\_CACHE\\_SIZE](#) 5
- #define [RCPP\\_HASH\\_CACHE\\_INITIAL\\_SIZE](#) 1024

## Functions

- SEXP [get\\_string\\_elt](#) (SEXP x, R\_xlen\_t i)
- const char \* [char\\_get\\_string\\_elt](#) (SEXP x, R\_xlen\_t i)
- void [set\\_string\\_elt](#) (SEXP x, R\_xlen\_t i, SEXP value)
- void [char\\_set\\_string\\_elt](#) (SEXP x, R\_xlen\_t i, const char \*value)
- SEXP \* [get\\_string\\_ptr](#) (SEXP x)
- SEXP [get\\_vector\\_elt](#) (SEXP x, R\_xlen\_t i)
- void [set\\_vector\\_elt](#) (SEXP x, R\_xlen\_t i, SEXP value)
- SEXP \* [get\\_vector\\_ptr](#) (SEXP x)
- void \* [dataptr](#) (SEXP x)
- const char \* [char\\_nocheck](#) (SEXP x)
- attribute\_hidden void [Rcpp::Rcpp\\_precious\\_init](#) ()
- attribute\_hidden void [Rcpp::Rcpp\\_precious\\_teardown](#) ()
- attribute\_hidden SEXP [Rcpp::Rcpp\\_precious\\_preserve](#) (SEXP object)
- attribute\_hidden void [Rcpp::Rcpp\\_precious\\_remove](#) (SEXP token)
- SEXP [get\\_rcpp\\_cache](#) ()
- attribute\_hidden SEXP [Rcpp::internal::get\\_Rcpp\\_namespace](#) ()
- SEXP [rcpp\\_get\\_stack\\_trace](#) ()
- SEXP [rcpp\\_set\\_stack\\_trace](#) (SEXP e)
- SEXP [set\\_error\\_occured](#) (SEXP cache, SEXP e)
- SEXP [set\\_current\\_error](#) (SEXP cache, SEXP e)
- SEXP [init\\_Rcpp\\_cache](#) ()
- SEXP [reset\\_current\\_error](#) ()
- int [error\\_occured](#) ()
- SEXP [rcpp\\_error\\_recorder](#) (SEXP e)
- SEXP [rcpp\\_get\\_current\\_error](#) ()
- int \* [get\\_cache](#) (int m)

## Variables

- static bool [Rcpp\\_cache\\_know](#) = false
- static SEXP [Rcpp\\_cache](#) = R\_NilValue
- static SEXP [Rcpp::Rcpp\\_precious](#) = R\_NilValue

## 7.397.1 Macro Definition Documentation

### 7.397.1.1 COMPILING\_RCPP

```
#define COMPILING_RCPP
```

Definition at line 22 of file barrier.cpp.

### 7.397.1.2 RCPP\_CACHE\_SIZE

```
#define RCPP_CACHE_SIZE 5
```

Definition at line 85 of file barrier.cpp.

### 7.397.1.3 RCPP\_HASH\_CACHE\_INDEX

```
#define RCPP_HASH_CACHE_INDEX 4
```

Definition at line 84 of file barrier.cpp.

### 7.397.1.4 RCPP\_HASH\_CACHE\_INITIAL\_SIZE

```
#define RCPP_HASH_CACHE_INITIAL_SIZE 1024
```

Definition at line 88 of file barrier.cpp.

### 7.397.1.5 USE\_RINTERNALS

```
#define USE_RINTERNALS
```

Definition at line 24 of file barrier.cpp.

## 7.397.2 Function Documentation

### 7.397.2.1 char\_get\_string\_elt()

```
const char* char_get_string_elt (
    SEXP x,
    R_xlen_t i ) [inline]
```

Definition at line 37 of file barrier.cpp.

References GET\_CALLABLE.

Referenced by registerFunctions().

### 7.397.2.2 char\_nocheck()

```
const char* char_nocheck (
    SEXP x ) [inline]
```

Definition at line 77 of file barrier.cpp.

References GET\_CALLABLE.

Referenced by registerFunctions(), and Rcpp::String::setBuffer().

### 7.397.2.3 char\_set\_string\_elt()

```
void char_set_string_elt (
    SEXP x,
    R_xlen_t i,
    const char * value ) [inline]
```

Definition at line 47 of file barrier.cpp.

References GET\_CALLABLE.

Referenced by registerFunctions().

### 7.397.2.4 dataptr()

```
void* dataptr (
    SEXP x ) [inline]
```

Definition at line 72 of file barrier.cpp.

References GET\_CALLABLE.

Referenced by registerFunctions().

### 7.397.2.5 error\_occured()

```
int error_occured ( )
```

Definition at line 213 of file barrier.cpp.

References get\_rcpp\_cache().

Here is the call graph for this function:



### 7.397.2.6 get\_cache()

```
int* get_cache (
    int m )
```

Definition at line 237 of file barrier.cpp.

References get\_rcpp\_cache(), and RCPP\_HASH\_CACHE\_INDEX.

Here is the call graph for this function:



### 7.397.2.7 `get_rcpp_cache()`

```
SEXP get_rcpp_cache ( )
```

Definition at line 133 of file barrier.cpp.

References `Rcpp_cache`, and `Rcpp_cache_know`.

Referenced by `error_occured()`, `get_cache()`, `rcpp_error_recorder()`, `rcpp_get_current_error()`, `rcpp_get_stack_trace()`, `rcpp_set_stack_trace()`, and `reset_current_error()`.

### 7.397.2.8 `get_string_elt()`

```
SEXP get_string_elt (
    SEXP x,
    R_xlen_t i ) [inline]
```

Definition at line 32 of file barrier.cpp.

References `GET_CALLABLE`.

Referenced by `registerFunctions()`.

### 7.397.2.9 `get_string_ptr()`

```
SEXP* get_string_ptr (
    SEXP x ) [inline]
```

Definition at line 52 of file barrier.cpp.

References `GET_CALLABLE`.

Referenced by `Rcpp::sugar::get_const_begin()`, and `registerFunctions()`.

### 7.397.2.10 `get_vector_elt()`

```
SEXP get_vector_elt (
    SEXP x,
    R_xlen_t i ) [inline]
```

Definition at line 57 of file barrier.cpp.

References `GET_CALLABLE`.

Referenced by `registerFunctions()`.



### 7.397.2.11 get\_vector\_ptr()

```
SEXP* get_vector_ptr (
    SEXP x ) [inline]
```

Definition at line 67 of file barrier.cpp.

References GET\_CALLABLE.

Referenced by registerFunctions().

### 7.397.2.12 init\_Rcpp\_cache()

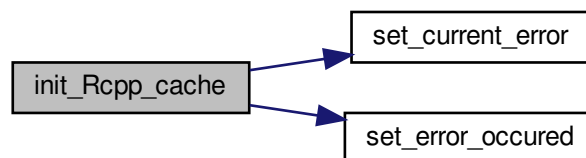
```
SEXP init_Rcpp_cache ( )
```

Definition at line 177 of file barrier.cpp.

References RCPP\_CACHE\_SIZE, RCPP\_HASH\_CACHE\_INDEX, RCPP\_HASH\_CACHE\_INITIAL\_SIZE, set\_↔current\_error(), and set\_error\_occured().

Referenced by R\_init\_Rcpp().

Here is the call graph for this function:



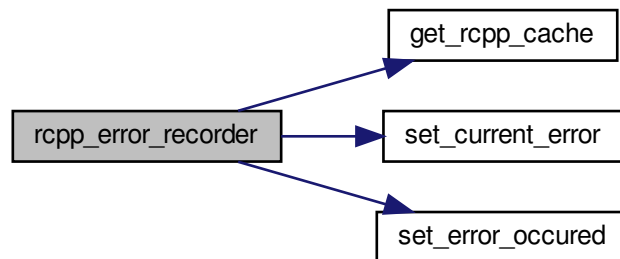
### 7.397.2.13 rcpp\_error\_recorder()

```
SEXP rcpp_error_recorder (  
    SEXP e )
```

Definition at line 219 of file barrier.cpp.

References `get_rcpp_cache()`, `set_current_error()`, and `set_error_occured()`.

Here is the call graph for this function:



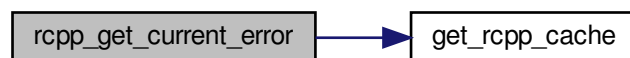
### 7.397.2.14 rcpp\_get\_current\_error()

```
SEXP rcpp_get_current_error ( )
```

Definition at line 232 of file barrier.cpp.

References `get_rcpp_cache()`.

Here is the call graph for this function:



### 7.397.2.15 rcpp\_get\_stack\_trace()

```
SEXP rcpp_get_stack_trace ( )
```

Definition at line 157 of file barrier.cpp.

References `get_rcpp_cache()`.

Here is the call graph for this function:



### 7.397.2.16 rcpp\_set\_stack\_trace()

```
SEXP rcpp_set_stack_trace (
    SEXP e )
```

Definition at line 162 of file barrier.cpp.

References `get_rcpp_cache()`.

Here is the call graph for this function:



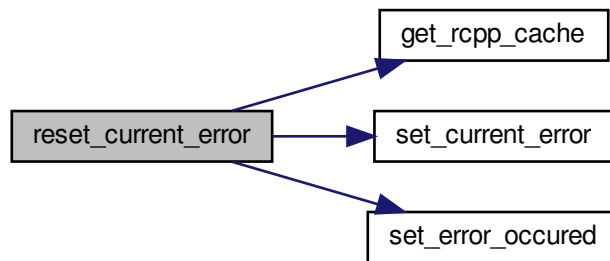
### 7.397.2.17 reset\_current\_error()

```
SEXP reset_current_error ( )
```

Definition at line 197 of file barrier.cpp.

References `get_rcpp_cache()`, `set_current_error()`, and `set_error_occured()`.

Here is the call graph for this function:



### 7.397.2.18 set\_current\_error()

```
SEXP set_current_error (
    SEXP cache,
    SEXP e )
```

Definition at line 172 of file barrier.cpp.

Referenced by `init_Rcpp_cache()`, `rcpp_error_recorder()`, and `reset_current_error()`.

### 7.397.2.19 set\_error\_occured()

```
SEXP set_error_occured (
    SEXP cache,
    SEXP e )
```

Definition at line 167 of file barrier.cpp.

Referenced by `init_Rcpp_cache()`, `rcpp_error_recorder()`, and `reset_current_error()`.

### 7.397.2.20 set\_string\_elt()

```
void set_string_elt (
    SEXP x,
    R_xlen_t i,
    SEXP value ) [inline]
```

Definition at line 42 of file barrier.cpp.

References GET\_CALLABLE.

Referenced by registerFunctions().

### 7.397.2.21 set\_vector\_elt()

```
void set_vector_elt (
    SEXP x,
    R_xlen_t i,
    SEXP value ) [inline]
```

Definition at line 62 of file barrier.cpp.

References GET\_CALLABLE.

Referenced by registerFunctions().

## 7.397.3 Variable Documentation

### 7.397.3.1 Rcpp\_cache

```
SEXP Rcpp_cache = R_NilValue [static]
```

Definition at line 82 of file barrier.cpp.

Referenced by get\_rcpp\_cache().

### 7.397.3.2 Rcpp\_cache\_know

```
bool Rcpp_cache_know = false [static]
```

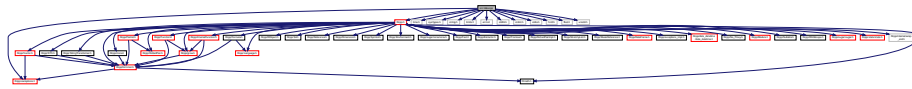
Definition at line 81 of file barrier.cpp.

Referenced by get\_rcpp\_cache().

## 7.398 src/date.cpp File Reference

```
#include <Rcpp.h>
#include <time.h>
#include <Rcpp/exceptions.h>
#include "sys/types.h"
#include "string.h"
#include "limits.h"
#include <errno.h>
#include "stdlib.h"
#include "stdint.h"
#include "stdio.h"
#include "fcntl.h"
#include "float.h"
#include <unistd.h>
```

Include dependency graph for date.cpp:



### Classes

- struct [Rcpp::tzhead](#)
- struct [Rcpp::tinfo](#)
- struct [Rcpp::lsinfo](#)
- struct [Rcpp::state](#)
- struct [Rcpp::rule](#)

### Namespaces

- [Rcpp](#)  
*Rcpp API.*

### Macros

- #define [COMPILING\\_RCPP](#)
- #define [isleap](#)(y) (((y) % 4) == 0 && ((y) % 100) != 0) || ((y) % 400) == 0)
- #define [days\\_in\\_year](#)(year) (isleap(year) ? 366 : 365)
- #define [\\_NO\\_OLDNAMES](#) /\* avoid tznames \*/
- #define [EOVERFLOW](#) 79
- #define [TYPE\\_BIT](#)(type) (sizeof (type) \* CHAR\_BIT)
- #define [TYPE\\_SIGNED](#)(type) (((type) - 1) < 0)
- #define [TYPE\\_INTEGRAL](#)(type) (((type) 0.5) != 0.5)
- #define [TWOES\\_COMPLEMENT](#)(t) ((t) ~ (t) 0 < 0)
- #define [GRANDPARENTED](#) "Local time zone must be set--see zic manual page"
- #define [YEARS\\_PERRPEAT](#) 400 /\* years before a Gregorian repeat \*/

- #define [AVGSECSPERYEAR](#) 31556952L
- #define [SECSPERREPEAT](#) ((int\_fast64\_t) [YEASPERREPEAT](#) \* (int\_fast64\_t) [AVGSECSPERYEAR](#))
- #define [SECSPERREPEAT\\_BITS](#) 34 /\* ceil(log2(SECSPERREPEAT)) \*/
- #define [is\\_digit](#)(c) ((unsigned)(c) - '0' <= 9)
- #define [INITIALIZE](#)(x) (x = 0)
- #define [MAXVAL](#)(t, b)
- #define [MINVAL](#)(t, b) ((t) (TYPE\_SIGNED(t) ? - [TWO\\_S\\_COMPLEMENT](#)(t) - [MAXVAL](#)(t, b) : 0))
- #define [TZFILE\\_H](#)
- #define [TZDIR](#) "/usr/local/etc/zoneinfo" /\* Time zone object file directory \*/
- #define [TZDEFAULT](#) "localtime"
- #define [TZDEFRULES](#) "America/New\_York"
- #define [TZ\\_MAGIC](#) "TZif"
- #define [TZ\\_MAX\\_TIMES](#) 1200
- #define [TZ\\_MAX\\_TYPES](#) 256 /\* Limited by what (unsigned char)'s can hold \*/
- #define [TZ\\_MAX\\_CHARS](#) 100 /\* Maximum number of abbreviation characters \*/
- #define [TZ\\_MAX\\_LEAPS](#) 50 /\* Maximum number of leap second corrections \*/
- #define [SECSPERMIN](#) 60
- #define [MINSPERHOUR](#) 60
- #define [HOURSPERDAY](#) 24
- #define [DAYSPERWEEK](#) 7
- #define [DAYSPERNYEAR](#) 365
- #define [DAYSPERLYEAR](#) 366
- #define [SECSPERHOUR](#) ([SECSPERMIN](#) \* [MINSPERHOUR](#))
- #define [SECSPERDAY](#) ((int\_fast32\_t) [SECSPERHOUR](#) \* [HOURSPERDAY](#))
- #define [MONSPERYEAR](#) 12
- #define [TM\\_SUNDAY](#) 0
- #define [TM\\_MONDAY](#) 1
- #define [TM\\_TUESDAY](#) 2
- #define [TM\\_WEDNESDAY](#) 3
- #define [TM\\_THURSDAY](#) 4
- #define [TM\\_FRIDAY](#) 5
- #define [TM\\_SATURDAY](#) 6
- #define [TM\\_JANUARY](#) 0
- #define [TM\\_FEBRUARY](#) 1
- #define [TM\\_MARCH](#) 2
- #define [TM\\_APRIL](#) 3
- #define [TM\\_MAY](#) 4
- #define [TM\\_JUNE](#) 5
- #define [TM\\_JULY](#) 6
- #define [TM\\_AUGUST](#) 7
- #define [TM\\_SEPTEMBER](#) 8
- #define [TM\\_OCTOBER](#) 9
- #define [TM\\_NOVEMBER](#) 10
- #define [TM\\_DECEMBER](#) 11
- #define [TM\\_YEAR\\_BASE](#) 1900
- #define [EPOCH\\_YEAR](#) 1970
- #define [EPOCH\\_WDAY](#) [TM\\_THURSDAY](#)
- #define [isleap](#)(y) (((y) % 4) == 0 && ((y) % 100) != 0 || ((y) % 400) == 0))
- #define [isleap\\_sum](#)(a, b) [isleap](#)((a) % 400 + (b) % 400)
- #define [OPEN\\_MODE](#) O\_RDONLY
- #define [TZDEFRULESTRING](#) ",M4.1.0,M10.5.0"

- #define `BIGGEST(a, b)` `((a) > (b)) ? (a) : (b)`
- #define `MY_TZNAME_MAX` 255
- #define `JULIAN_DAY` 0 /\* Jn - Julian day \*/
- #define `DAY_OF_YEAR` 1 /\* n - day of year \*/
- #define `MONTH_NTH_DAY_OF_WEEK` 2 /\* Mm.n.d - month, week, day of week \*/
- #define `gmtptr` (&gmtmem)

## Functions

- attribute\_hidden double `Rcpp::mktime00` (struct tm &tm)
- static int `Rcpp::tzparse` (const char \*name, struct state \*sp, int lastditch)
- static int `Rcpp::typesequiv` (const struct state \*sp, int a, int b)
- static const char \* `Rcpp::getsecs` (const char \*strp, int\_fast32\_t \*secsp)
- static const char \* `Rcpp::getnum` (const char \*strp, int \*const nump, const int min, const int max)
- static const char \* `Rcpp::getrule` (const char \*strp, struct rule \*const rulep)
- static int\_fast32\_t `Rcpp::transtime` (int year, const struct rule \*rulep, int\_fast32\_t offset)
- static struct tm \* `Rcpp::timesub` (const time\_t \*timep, int\_fast32\_t offset, const struct state \*sp, struct tm \*tmp)
- static int `Rcpp::leaps_thru_end_of` (const int y)
- static int `Rcpp::increment_overflow` (int \*const ip, int j)
- static int `Rcpp::increment_overflow_time` (time\_t \*tp, int\_fast32\_t j)
- static int\_fast32\_t `Rcpp::detzcode` (const char \*const codep)
- static int\_fast64\_t `Rcpp::detzcode64` (const char \*const codep)
- static int `Rcpp::differ_by_repeat` (const time\_t t1, const time\_t t0)
- static const char \* `Rcpp::getzname` (const char \*strp)
- static const char \* `Rcpp::getqzname` (const char \*strp, const int delim)
- static const char \* `Rcpp::getoffset` (const char \*strp, int\_fast32\_t \*const offsetp)
- static int `Rcpp::tzload` (const char \*name, struct state \*const sp, const int doextend)
- static void `Rcpp::gmtload` (struct state \*const sp)
- static struct tm \* `Rcpp::gmtsub` (const time\_t \*const timep, const int\_fast32\_t offset, struct tm \*const tmp)
- attribute\_hidden struct tm \* `Rcpp::gmtime_` (const time\_t \*const x)

## Variables

- static time\_t const `Rcpp::time_t_min` = `MINVAL(time_t, TYPE_BIT(time_t))`
- static time\_t const `Rcpp::time_t_max` = `MAXVAL(time_t, TYPE_BIT(time_t))`
- static const char `Rcpp::gmt` [] = "GMT"
- static const int `Rcpp::mon_lengths` [2][`MONSPERYEAR`]
- static const int `Rcpp::year_lengths` [2]
- static int `Rcpp::gmt_is_set`
- static struct state `Rcpp::gmtmem`
- static struct tm `Rcpp::tm`

### 7.398.1 Macro Definition Documentation



### 7.398.1.1 `_NO_OLDNAMES`

```
#define _NO_OLDNAMES /* avoid tznames */
```

Definition at line 120 of file date.cpp.

### 7.398.1.2 `AVGSECSPERYEAR`

```
#define AVGSECSPERYEAR 31556952L
```

Definition at line 145 of file date.cpp.

### 7.398.1.3 `BIGGEST`

```
#define BIGGEST(  
    a,  
    b ) ((a) > (b)) ? (a) : (b)
```

Definition at line 371 of file date.cpp.

### 7.398.1.4 `COMPILING_RCPP`

```
#define COMPILING_RCPP
```

Definition at line 28 of file date.cpp.

### 7.398.1.5 `DAY_OF_YEAR`

```
#define DAY_OF_YEAR 1 /* n - day of year */
```

Definition at line 417 of file date.cpp.

### 7.398.1.6 days\_in\_year

```
#define days_in_year(  
    year ) (isleap(year) ? 366 : 365)
```

### 7.398.1.7 DAYSPERLYEAR

```
#define DAYSPERLYEAR 366
```

Definition at line 296 of file date.cpp.

### 7.398.1.8 DAYSPERNYEAR

```
#define DAYSPERNYEAR 365
```

Definition at line 295 of file date.cpp.

### 7.398.1.9 DAYSPERWEEK

```
#define DAYSPERWEEK 7
```

Definition at line 294 of file date.cpp.

### 7.398.1.10 EOVERFLOW

```
#define EOVERFLOW 79
```

Definition at line 126 of file date.cpp.

### 7.398.1.11 EPOCH\_WDAY

```
#define EPOCH_WDAY TM_THURSDAY
```

Definition at line 325 of file date.cpp.

### 7.398.1.12 EPOCH\_YEAR

```
#define EPOCH_YEAR 1970
```

Definition at line 324 of file date.cpp.

### 7.398.1.13 gmtptr

```
#define gmtptr (&gmtmem)
```

Definition at line 434 of file date.cpp.

### 7.398.1.14 GRANDPARENTED

```
#define GRANDPARENTED "Local time zone must be set--see zic manual page"
```

Definition at line 143 of file date.cpp.

### 7.398.1.15 HOURS PER DAY

```
#define HOURS PER DAY 24
```

Definition at line 293 of file date.cpp.

### 7.398.1.16 INITIALIZE

```
#define INITIALIZE(  
    x ) (x = 0)
```

Definition at line 149 of file date.cpp.

**7.398.1.17 is\_digit**

```
#define is_digit(  
    c ) ((unsigned)(c) - '0' <= 9)
```

Definition at line 148 of file date.cpp.

**7.398.1.18 isleap [1/2]**

```
#define isleap(  
    y ) (((y) % 4) == 0 && ((y) % 100) != 0) || ((y) % 400) == 0)
```

Definition at line 327 of file date.cpp.

**7.398.1.19 isleap [2/2]**

```
#define isleap(  
    y ) (((y) % 4) == 0 && ((y) % 100) != 0 || ((y) % 400) == 0))
```

Definition at line 327 of file date.cpp.

**7.398.1.20 isleap\_sum**

```
#define isleap_sum(  
    a,  
    b ) isleap((a) % 400 + (b) % 400)
```

Definition at line 341 of file date.cpp.

**7.398.1.21 JULIAN\_DAY**

```
#define JULIAN_DAY 0 /* Jn - Julian day */
```

Definition at line 416 of file date.cpp.

### 7.398.1.22 MAXVAL

```
#define MAXVAL(  
    t,  
    b )
```

**Value:**

```
((t) (((t) 1 << ((b) - 1 - TYPE_SIGNED(t)))  
- 1 + ((t) 1 << ((b) - 1 - TYPE_SIGNED(t)))))
```

Definition at line 154 of file date.cpp.

### 7.398.1.23 MINSPERHOUR

```
#define MINSPERHOUR 60
```

Definition at line 292 of file date.cpp.

### 7.398.1.24 MINVAL

```
#define MINVAL(  
    t,  
    b ) ((t) (TYPE_SIGNED(t) ? - TWOS_COMPLEMENT(t) - MAXVAL(t, b) : 0))
```

Definition at line 157 of file date.cpp.

### 7.398.1.25 MONSPERYEAR

```
#define MONSPERYEAR 12
```

Definition at line 299 of file date.cpp.

### 7.398.1.26 MONTH\_NTH\_DAY\_OF\_WEEK

```
#define MONTH_NTH_DAY_OF_WEEK 2 /* Mm.n.d - month, week, day of week */
```

Definition at line 418 of file date.cpp.

### 7.398.1.27 MY\_TZNAME\_MAX

```
#define MY_TZNAME_MAX 255
```

Definition at line 377 of file date.cpp.

### 7.398.1.28 OPEN\_MODE

```
#define OPEN_MODE O_RDONLY
```

Definition at line 355 of file date.cpp.

### 7.398.1.29 SECSPERDAY

```
#define SECSPERDAY ((int_fast32_t) SECSPERHOUR * HOURSPERDAY)
```

Definition at line 298 of file date.cpp.

### 7.398.1.30 SECSPERHOUR

```
#define SECSPERHOUR (SECSPERMIN * MINSPERHOUR)
```

Definition at line 297 of file date.cpp.

### 7.398.1.31 SECSPERMIN

```
#define SECSPERMIN 60
```

Definition at line 291 of file date.cpp.

### 7.398.1.32 SECSPERREPEAT

```
#define SECSPERREPEAT ((int_fast64_t) YEARSPPERREPEAT * (int_fast64_t) AVGSECSPERYEAR)
```

Definition at line 146 of file date.cpp.

### 7.398.1.33 SECSPERREPEAT\_BITS

```
#define SECSPERREPEAT_BITS 34 /* ceil(log2(SECSPERREPEAT)) */
```

Definition at line 147 of file date.cpp.

### 7.398.1.34 TM\_APRIL

```
#define TM_APRIL 3
```

Definition at line 312 of file date.cpp.

### 7.398.1.35 TM\_AUGUST

```
#define TM_AUGUST 7
```

Definition at line 316 of file date.cpp.

### 7.398.1.36 TM\_DECEMBER

```
#define TM_DECEMBER 11
```

Definition at line 320 of file date.cpp.

### 7.398.1.37 TM\_FEBRUARY

```
#define TM_FEBRUARY 1
```

Definition at line 310 of file date.cpp.

### 7.398.1.38 TM\_FRIDAY

```
#define TM_FRIDAY 5
```

Definition at line 306 of file date.cpp.

**7.398.1.39 TM\_JANUARY**

```
#define TM_JANUARY 0
```

Definition at line 309 of file date.cpp.

**7.398.1.40 TM\_JULY**

```
#define TM_JULY 6
```

Definition at line 315 of file date.cpp.

**7.398.1.41 TM\_JUNE**

```
#define TM_JUNE 5
```

Definition at line 314 of file date.cpp.

**7.398.1.42 TM\_MARCH**

```
#define TM_MARCH 2
```

Definition at line 311 of file date.cpp.

**7.398.1.43 TM\_MAY**

```
#define TM_MAY 4
```

Definition at line 313 of file date.cpp.

**7.398.1.44 TM\_MONDAY**

```
#define TM_MONDAY 1
```

Definition at line 302 of file date.cpp.



**7.398.1.45 TM\_NOVEMBER**

```
#define TM_NOVEMBER 10
```

Definition at line 319 of file date.cpp.

**7.398.1.46 TM\_OCTOBER**

```
#define TM_OCTOBER 9
```

Definition at line 318 of file date.cpp.

**7.398.1.47 TM\_SATURDAY**

```
#define TM_SATURDAY 6
```

Definition at line 307 of file date.cpp.

**7.398.1.48 TM\_SEPTEMBER**

```
#define TM_SEPTEMBER 8
```

Definition at line 317 of file date.cpp.

**7.398.1.49 TM\_SUNDAY**

```
#define TM_SUNDAY 0
```

Definition at line 301 of file date.cpp.

**7.398.1.50 TM\_THURSDAY**

```
#define TM_THURSDAY 4
```

Definition at line 305 of file date.cpp.

**7.398.1.51 TM\_TUESDAY**

```
#define TM_TUESDAY 2
```

Definition at line 303 of file date.cpp.

**7.398.1.52 TM\_WEDNESDAY**

```
#define TM_WEDNESDAY 3
```

Definition at line 304 of file date.cpp.

**7.398.1.53 TM\_YEAR\_BASE**

```
#define TM_YEAR_BASE 1900
```

Definition at line 322 of file date.cpp.

**7.398.1.54 TWOS\_COMPLEMENT**

```
#define TWOS_COMPLEMENT(  
    t ) ((t) ~ (t) 0 < 0)
```

Definition at line 142 of file date.cpp.

**7.398.1.55 TYPE\_BIT**

```
#define TYPE_BIT(  
    type ) (sizeof (type) * CHAR_BIT)
```

Definition at line 139 of file date.cpp.

### 7.398.1.56 TYPE\_INTEGRAL

```
#define TYPE_INTEGRAL(  
    type ) (((type) 0.5) != 0.5)
```

Definition at line 141 of file date.cpp.

### 7.398.1.57 TYPE\_SIGNED

```
#define TYPE_SIGNED(  
    type ) (((type) -1) < 0)
```

Definition at line 140 of file date.cpp.

### 7.398.1.58 TZ\_MAGIC

```
#define TZ_MAGIC "TZif"
```

Definition at line 203 of file date.cpp.

### 7.398.1.59 TZ\_MAX\_CHARS

```
#define TZ_MAX_CHARS 100 /* Maximum number of abbreviation characters */
```

Definition at line 283 of file date.cpp.

### 7.398.1.60 TZ\_MAX\_LEAPS

```
#define TZ_MAX_LEAPS 50 /* Maximum number of leap second corrections */
```

Definition at line 288 of file date.cpp.

### 7.398.1.61 TZ\_MAX\_TIMES

```
#define TZ_MAX_TIMES 1200
```

Definition at line 265 of file date.cpp.

### 7.398.1.62 TZ\_MAX\_TYPES

```
#define TZ_MAX_TYPES 256 /* Limited by what (unsigned char)'s can hold */
```

Definition at line 270 of file date.cpp.

### 7.398.1.63 TZDEFAULT

```
#define TZDEFAULT "localtime"
```

Definition at line 192 of file date.cpp.

### 7.398.1.64 TZDEFRULES

```
#define TZDEFRULES "America/New_York"
```

Definition at line 196 of file date.cpp.

### 7.398.1.65 TZDEFRULESTRING

```
#define TZDEFRULESTRING ",M4.1.1.0,M10.5.0"
```

Definition at line 368 of file date.cpp.

### 7.398.1.66 TZDIR

```
#define TZDIR "/usr/local/etc/zoneinfo" /* Time zone object file directory */
```

Definition at line 188 of file date.cpp.

### 7.398.1.67 TZFILE\_H

```
#define TZFILE_H
```

Definition at line 168 of file date.cpp.

### 7.398.1.68 YEARSPPERREPEAT

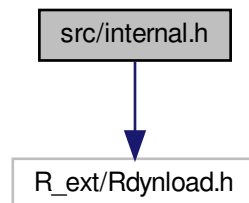
```
#define YEARSPPERREPEAT 400 /* years before a Gregorian repeat */
```

Definition at line 144 of file date.cpp.

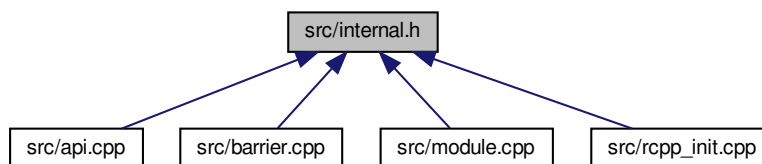
## 7.399 src/internal.h File Reference

```
#include <R_ext/Rdynload.h>
```

Include dependency graph for internal.h:



This graph shows which files directly or indirectly include this file:



## Macros

- #define [CALLFUN\\_0](#)(name) SEXP name()
- #define [CALLFUN\\_1](#)(name) SEXP name(SEXP)
- #define [CALLFUN\\_2](#)(name) SEXP name(SEXP,SEXP)
- #define [CALLFUN\\_3](#)(name) SEXP name(SEXP,SEXP,SEXP)
- #define [CALLFUN\\_4](#)(name) SEXP name(SEXP,SEXP,SEXP,SEXP)
- #define [CALLFUN\\_5](#)(name) SEXP name(SEXP,SEXP,SEXP,SEXP,SEXP)
- #define [EXTFUN](#)(name) SEXP name(SEXP)
- #define [MAX\\_ARGS](#) 65
- #define [UNPACK\\_EXTERNAL\\_ARGS](#)(\_\_CARGS\_\_, \_\_P\_\_)
- #define [RCPP\\_FUN\\_1](#)(\_\_OUT\_\_, \_\_NAME\_\_, \_\_0)
- #define [RCPP\\_FUN\\_2](#)(\_\_OUT\_\_, \_\_NAME\_\_, \_\_0, \_\_1)
- #define [RCPP\\_FUN\\_3](#)(\_\_OUT\_\_, \_\_NAME\_\_, \_\_0, \_\_1, \_\_2)
- #define [RCPP\\_FUN\\_4](#)(\_\_OUT\_\_, \_\_NAME\_\_, \_\_0, \_\_1, \_\_2, \_\_3)

## Functions

- SEXP [get\\_Rcpp\\_protection\\_stack](#) ()
- [CALLFUN\\_1](#) (as\_character\_externalptr)
- [CALLFUN\\_1](#) (Class\_\_name)
- [CALLFUN\\_1](#) (Class\_\_has\_default\_constructor)
- [CALLFUN\\_1](#) (CppClass\_\_complete)
- [CALLFUN\\_1](#) (CppClass\_\_methods)
- [CALLFUN\\_1](#) (Module\_\_classes\_info)
- [CALLFUN\\_1](#) (Module\_\_complete)
- [CALLFUN\\_1](#) (Module\_\_functions\_arity)
- [CALLFUN\\_1](#) (Module\_\_functions\_names)
- [CALLFUN\\_2](#) (Module\_\_get\_class)
- [CALLFUN\\_2](#) (Module\_\_has\_class)
- [CALLFUN\\_2](#) (Module\_\_has\_function)
- [CALLFUN\\_2](#) (Module\_\_get\_function)
- [CALLFUN\\_1](#) (Module\_\_name)
- [CALLFUN\\_2](#) (CppClass\_\_finalize)
- [CALLFUN\\_0](#) (get\_rcpp\_cache)
- [CALLFUN\\_0](#) (init\_Rcpp\_cache)
- [CALLFUN\\_1](#) (rcpp\_error\_recorder)
- [CALLFUN\\_3](#) (CppClass\_\_get)
- [CALLFUN\\_4](#) (CppClass\_\_set)
- [CALLFUN\\_0](#) (rcpp\_capabilities)
- [CALLFUN\\_0](#) (rcpp\_can\_use\_cxx0x)
- [CALLFUN\\_0](#) (rcpp\_can\_use\_cxx11)
- [CALLFUN\\_0](#) (getRcppVersionStrings)
- [EXTFUN](#) (CppClass\_\_invoke)
- [EXTFUN](#) (CppClass\_\_invoke\_void)
- [EXTFUN](#) (CppClass\_\_invoke\_notvoid)
- [EXTFUN](#) (InternalFunction\_\_invoke)
- [EXTFUN](#) (Module\_\_invoke)
- [EXTFUN](#) (class\_\_newInstance)
- [EXTFUN](#) (class\_\_dummyInstance)
- void [init\\_Rcpp\\_routines](#) (DllInfo \*)

## 7.399.1 Macro Definition Documentation

### 7.399.1.1 CALLFUN\_0

```
#define CALLFUN_0(  
    name ) SEXP name()
```

Definition at line 27 of file internal.h.

### 7.399.1.2 CALLFUN\_1

```
#define CALLFUN_1(  
    name ) SEXP name(SEXP)
```

Definition at line 28 of file internal.h.

### 7.399.1.3 CALLFUN\_2

```
#define CALLFUN_2(  
    name ) SEXP name(SEXP, SEXP)
```

Definition at line 29 of file internal.h.

### 7.399.1.4 CALLFUN\_3

```
#define CALLFUN_3(  
    name ) SEXP name(SEXP, SEXP, SEXP)
```

Definition at line 30 of file internal.h.

### 7.399.1.5 CALLFUN\_4

```
#define CALLFUN_4(  
    name ) SEXP name(SEXP, SEXP, SEXP, SEXP)
```

Definition at line 31 of file internal.h.

### 7.399.1.6 CALLFUN\_5

```
#define CALLFUN_5(
    name ) SEXP name (SEXP, SEXP, SEXP, SEXP, SEXP)
```

Definition at line 32 of file internal.h.

### 7.399.1.7 EXTFUN

```
#define EXTFUN(
    name ) SEXP name (SEXP)
```

Definition at line 33 of file internal.h.

### 7.399.1.8 MAX\_ARGS

```
#define MAX_ARGS 65
```

Definition at line 38 of file internal.h.

### 7.399.1.9 RCPP\_FUN\_1

```
#define RCPP_FUN_1(
    __OUT__,
    __NAME__,
    __0 )
```

**Value:**

```
__OUT__ RCPP_DECORATE (__NAME__) (__0);
SEXP __NAME__(SEXP x0) {
    SEXP res = R_NilValue;
    BEGIN_RCPP
    res = ::Rcpp::wrap(RCPP_DECORATE (__NAME__) (::Rcpp::internal::converter(x0))); \
    return res;
    END_RCPP
}
__OUT__ RCPP_DECORATE (__NAME__) (__0)
```

Definition at line 48 of file internal.h.



## 7.399.1.10 RCPP\_FUN\_2

```
#define RCPP_FUN_2(
    __OUT__,
    __NAME__,
    __0,
    __1 )
```

**Value:**

```
__OUT__ RCPP_DECORATE(__NAME__)(__0, __1);
SEXP __NAME__(SEXP x0, SEXP x1) {
    SEXP res = R_NilValue;
    BEGIN_RCPP
    res = ::Rcpp::wrap(RCPP_DECORATE(__NAME__)(::Rcpp::internal::converter(x0), \
                                                ::Rcpp::internal::converter(x1)));
    return res;
    END_RCPP
}
__OUT__ RCPP_DECORATE(__NAME__)(__0, __1)
```

Definition at line 60 of file internal.h.

## 7.399.1.11 RCPP\_FUN\_3

```
#define RCPP_FUN_3(
    __OUT__,
    __NAME__,
    __0,
    __1,
    __2 )
```

**Value:**

```
__OUT__ RCPP_DECORATE(__NAME__)(__0, __1, __2);
SEXP __NAME__(SEXP x0, SEXP x1, SEXP x2) {
    SEXP res = R_NilValue;
    BEGIN_RCPP
    res = ::Rcpp::wrap(RCPP_DECORATE(__NAME__)(::Rcpp::internal::converter(x0), \
                                                ::Rcpp::internal::converter(x1), \
                                                ::Rcpp::internal::converter(x2)));
    return res;
    END_RCPP
}
__OUT__ RCPP_DECORATE(__NAME__)(__0, __1, __2)
```

Definition at line 72 of file internal.h.

### 7.399.1.12 RCPP\_FUN\_4

```
#define RCPP_FUN_4(
    __OUT__,
    __NAME__,
    __0__,
    __1__,
    __2__,
    __3__ )
```

**Value:**

```
__OUT__ RCPP_DECORATE(__NAME__)(__0__, __1__, __2__, __3__);
SEXP __NAME__(SEXP x0, SEXP x1, SEXP x2, SEXP x3) {
    SEXP res = R_NilValue;
    BEGIN_RCPP
        res = ::Rcpp::wrap( RCPP_DECORATE(__NAME__) (::Rcpp::internal::converter(x0), \
                                                    ::Rcpp::internal::converter(x1), \
                                                    ::Rcpp::internal::converter(x2), \
                                                    ::Rcpp::internal::converter(x3))); \
    return res;
    END_RCPP
}
__OUT__ RCPP_DECORATE(__NAME__)(__0__, __1__, __2__, __3)
```

Definition at line 85 of file internal.h.

### 7.399.1.13 UNPACK\_EXTERNAL\_ARGS

```
#define UNPACK_EXTERNAL_ARGS(
    __CARGS__,
    __P__ )
```

**Value:**

```
SEXP __CARGS__[MAX_ARGS];
int nargs = 0;
for (; nargs<MAX_ARGS; nargs++) {
    if (Rf_isNull(__P__)) break;
    __CARGS__[nargs] = CAR(__P__);
    __P__ = CDR(__P__);
}
```

Definition at line 39 of file internal.h.

## 7.399.2 Function Documentation

### 7.399.2.1 CALLFUN\_0() [1/6]

```
CALLFUN_0 (
    get_rcpp_cache )
```

**7.399.2.2 CALLFUN\_0()** [2/6]

```
CALLFUN_0 (
    getRcppVersionStrings )
```

**7.399.2.3 CALLFUN\_0()** [3/6]

```
CALLFUN_0 (
    init_Rcpp_cache )
```

**7.399.2.4 CALLFUN\_0()** [4/6]

```
CALLFUN_0 (
    rcpp_can_use_cxx0x )
```

**7.399.2.5 CALLFUN\_0()** [5/6]

```
CALLFUN_0 (
    rcpp_can_use_cxx11 )
```

**7.399.2.6 CALLFUN\_0()** [6/6]

```
CALLFUN_0 (
    rcpp_capabilities )
```

**7.399.2.7 CALLFUN\_1()** [1/11]

```
CALLFUN_1 (
    as_character_externalptr )
```

**7.399.2.8 CALLFUN\_1()** [2/11]

```
CALLFUN_1 (
    Class__has_default_constructor )
```

**7.399.2.9 CALLFUN\_1()** [3/11]

```
CALLFUN_1 (
    Class__name )
```

**7.399.2.10 CALLFUN\_1()** [4/11]

```
CALLFUN_1 (
    CppClass__complete )
```

**7.399.2.11 CALLFUN\_1()** [5/11]

```
CALLFUN_1 (
    CppClass__methods )
```

**7.399.2.12 CALLFUN\_1()** [6/11]

```
CALLFUN_1 (
    Module__classes_info )
```

**7.399.2.13 CALLFUN\_1()** [7/11]

```
CALLFUN_1 (
    Module__complete )
```

**7.399.2.14 CALLFUN\_1()** [8/11]

```
CALLFUN_1 (
    Module__functions_arity )
```

**7.399.2.15 CALLFUN\_1()** [9/11]

```
CALLFUN_1 (
    Module__functions_names )
```

**7.399.2.16 CALLFUN\_1()** [10/11]

```
CALLFUN_1 (
    Module__name )
```

**7.399.2.17 CALLFUN\_1()** [11/11]

```
CALLFUN_1 (
    rcpp_error_recorder )
```

**7.399.2.18 CALLFUN\_2()** [1/5]

```
CALLFUN_2 (
    CppObject__finalize )
```

**7.399.2.19 CALLFUN\_2()** [2/5]

```
CALLFUN_2 (
    Module__get_class )
```

**7.399.2.20 CALLFUN\_2()** [3/5]

```
CALLFUN_2 (
    Module__get_function )
```

**7.399.2.21 CALLFUN\_2()** [4/5]

```
CALLFUN_2 (
    Module__has_class )
```

**7.399.2.22 CALLFUN\_2()** [5/5]

```
CALLFUN_2 (
    Module__has_function )
```

**7.399.2.23 CALLFUN\_3()**

```
CALLFUN_3 (
    CppField__get )
```

**7.399.2.24 CALLFUN\_4()**

```
CALLFUN_4 (
    CppField__set )
```

**7.399.2.25 EXTFUN()** [1/7]

```
EXTFUN (
    class__dummyInstance )
```

**7.399.2.26** EXTFUN() [2/7]

```
EXTFUN (
    class__newInstance )
```

**7.399.2.27** EXTFUN() [3/7]

```
EXTFUN (
    CppMethod__invoke )
```

**7.399.2.28** EXTFUN() [4/7]

```
EXTFUN (
    CppMethod__invoke_notvoid )
```

**7.399.2.29** EXTFUN() [5/7]

```
EXTFUN (
    CppMethod__invoke_void )
```

**7.399.2.30** EXTFUN() [6/7]

```
EXTFUN (
    InternalFunction_invoke )
```

**7.399.2.31** EXTFUN() [7/7]

```
EXTFUN (
    Module__invoke )
```

### 7.399.2.32 `get_Rcpp_protection_stack()`

```
SEXP get_Rcpp_protection_stack ( )
```

### 7.399.2.33 `init_Rcpp_routines()`

```
void init_Rcpp_routines (
    DllInfo * info )
```

Definition at line 78 of file `rcpp_init.cpp`.

References `callEntries`, and `extEntries`.

Referenced by `R_init_Rcpp()`.

## 7.400 `src/module.cpp` File Reference

```
#include <Rcpp.h>
#include "internal.h"
Include dependency graph for module.cpp:
```



## Namespaces

- [Rcpp](#)  
*Rcpp API.*

## Macros

- `#define COMPILING_RCPP`
- `#define CHECK_DUMMY_OBJ(p) if (p == rcpp\_dummy\_pointer) throw Rcpp::not_initialized()`

## Typedefs

- typedef `Rcpp::XPtr< Rcpp::Module > XP_Module`
- typedef `Rcpp::XPtr< Rcpp::class_Base > XP_Class`
- typedef `Rcpp::XPtr< Rcpp::CppMethodBase > XP_Function`



## Functions

- [RCPP\\_FUN\\_1](#) (bool, [Class\\_\\_has\\_default\\_constructor](#), [XP\\_Class](#) cl)
- [RCPP\\_FUN\\_2](#) (SEXP, [Module\\_\\_get\\_function](#), [XP\\_Module](#) module, std::string fun)
- [RCPP\\_FUN\\_2](#) (bool, [Class\\_\\_has\\_method](#), [XP\\_Class](#) cl, std::string m)
- [RCPP\\_FUN\\_2](#) (bool, [Class\\_\\_has\\_property](#), [XP\\_Class](#) cl, std::string m)
- [RCPP\\_FUN\\_1](#) (std::string, [Class\\_\\_name](#), [XP\\_Class](#) cl)
- [RCPP\\_FUN\\_2](#) (bool, [Module\\_\\_has\\_function](#), [XP\\_Module](#) module, std::string met)
- [RCPP\\_FUN\\_2](#) (bool, [Module\\_\\_has\\_class](#), [XP\\_Module](#) module, std::string cl)
- [RCPP\\_FUN\\_2](#) ([Rcpp::CppClass](#), [Module\\_\\_get\\_class](#), [XP\\_Module](#) module, std::string cl)
- [RCPP\\_FUN\\_1](#) (bool, [CppObject\\_\\_needs\\_init](#), SEXP xp)
- [RCPP\\_FUN\\_1](#) ([Rcpp::CharacterVector](#), [CppClass\\_\\_methods](#), [XP\\_Class](#) cl)
- [RCPP\\_FUN\\_1](#) ([Rcpp::CharacterVector](#), [CppClass\\_\\_properties](#), [XP\\_Class](#) cl)
- [RCPP\\_FUN\\_1](#) ([Rcpp::List](#), [CppClass\\_\\_property\\_classes](#), [XP\\_Class](#) cl)
- [RCPP\\_FUN\\_1](#) ([Rcpp::IntegerVector](#), [CppClass\\_\\_methods\\_arity](#), [XP\\_Class](#) cl)
- [RCPP\\_FUN\\_1](#) ([Rcpp::LogicalVector](#), [CppClass\\_\\_methods\\_voidness](#), [XP\\_Class](#) cl)
- [RCPP\\_FUN\\_2](#) (bool, [CppClass\\_\\_property\\_is\\_readonly](#), [XP\\_Class](#) cl, std::string p)
- [RCPP\\_FUN\\_2](#) (std::string, [CppClass\\_\\_property\\_class](#), [XP\\_Class](#) cl, std::string p)
- [RCPP\\_FUN\\_1](#) ([Rcpp::IntegerVector](#), [Module\\_\\_functions\\_arity](#), [XP\\_Module](#) module)
- [RCPP\\_FUN\\_1](#) ([Rcpp::CharacterVector](#), [Module\\_\\_functions\\_names](#), [XP\\_Module](#) module)
- [RCPP\\_FUN\\_1](#) (std::string, [Module\\_\\_name](#), [XP\\_Module](#) module)
- [RCPP\\_FUN\\_1](#) ([Rcpp::List](#), [Module\\_\\_classes\\_info](#), [XP\\_Module](#) module)
- [RCPP\\_FUN\\_1](#) ([Rcpp::CharacterVector](#), [Module\\_\\_complete](#), [XP\\_Module](#) module)
- [RCPP\\_FUN\\_1](#) ([Rcpp::CharacterVector](#), [CppClass\\_\\_complete](#), [XP\\_Class](#) cl)
- [RCPP\\_FUN\\_3](#) (SEXP, [CppClass\\_\\_get](#), [XP\\_Class](#) cl, SEXP field\_xp, SEXP obj)
- [RCPP\\_FUN\\_4](#) (SEXP, [CppClass\\_\\_set](#), [XP\\_Class](#) cl, SEXP field\_xp, SEXP obj, SEXP value)
- [RCPP\\_FUN\\_2](#) (SEXP, [CppObject\\_\\_finalize](#), [XP\\_Class](#) cl, SEXP obj)
- SEXP [InternalFunction\\_invoke](#) (SEXP args)
- SEXP [Module\\_\\_invoke](#) (SEXP args)
- SEXP [class\\_\\_newInstance](#) (SEXP args)
- SEXP [class\\_\\_dummyInstance](#) (SEXP args)
- SEXP [CppMethod\\_invoke](#) (SEXP args)
- SEXP [CppMethod\\_invoke\\_void](#) (SEXP args)
- SEXP [CppMethod\\_invoke\\_notvoid](#) (SEXP args)
- [Rcpp::Module](#) \* [getCurrentScope](#) ()
- void [setCurrentScope](#) ([Rcpp::Module](#) \*scope)

## Variables

- SEXP [rcpp\\_dummy\\_pointer](#) = R\_NilValue
- static Module \* [Rcpp::current\\_scope](#)

### 7.400.1 Macro Definition Documentation

### 7.400.1.1 CHECK\_DUMMY\_OBJ

```
#define CHECK_DUMMY_OBJ(  
    p ) if (p == rcpp\_dummy\_pointer) throw Rcpp::not_initialized()
```

Definition at line 149 of file module.cpp.

### 7.400.1.2 COMPILING\_RCPP

```
#define COMPILING_RCPP
```

Definition at line 22 of file module.cpp.

## 7.400.2 Typedef Documentation

### 7.400.2.1 XP\_Class

```
typedef Rcpp::XPtr<Rcpp::class\_Base> XP_Class
```

Definition at line 28 of file module.cpp.

### 7.400.2.2 XP\_Function

```
typedef Rcpp::XPtr<Rcpp::CppFunctionBase> XP_Function
```

Definition at line 29 of file module.cpp.

### 7.400.2.3 XP\_Module

```
typedef Rcpp::XPtr<Rcpp::Module> XP_Module
```

Definition at line 27 of file module.cpp.

## 7.400.3 Function Documentation

### 7.400.3.1 class\_\_dummyInstance()

```
SEXP class__dummyInstance (
    SEXP args )
```

Definition at line 152 of file module.cpp.

References rcpp\_dummy\_pointer.

### 7.400.3.2 class\_\_newInstance()

```
SEXP class__newInstance (
    SEXP args )
```

Definition at line 137 of file module.cpp.

References UNPACK\_EXTERNAL\_ARGS.

### 7.400.3.3 CppMethod\_\_invoke()

```
SEXP CppMethod__invoke (
    SEXP args )
```

Definition at line 166 of file module.cpp.

References CHECK\_DUMMY\_OBJ, and UNPACK\_EXTERNAL\_ARGS.

### 7.400.3.4 CppMethod\_\_invoke\_notvoid()

```
SEXP CppMethod__invoke_notvoid (
    SEXP args )
```

Definition at line 204 of file module.cpp.

References CHECK\_DUMMY\_OBJ, and UNPACK\_EXTERNAL\_ARGS.

#### 7.400.3.5 CppMethod\_\_invoke\_void()

```
SEXP CppMethod__invoke_void (
    SEXP args )
```

Definition at line 185 of file module.cpp.

References CHECK\_DUMMY\_OBJ, and UNPACK\_EXTERNAL\_ARGS.

#### 7.400.3.6 getCurrentScope()

```
Rcpp::Module* getCurrentScope ( )
```

Definition at line 227 of file module.cpp.

References Rcpp::current\_scope.

#### 7.400.3.7 InternalFunction\_invoke()

```
SEXP InternalFunction_invoke (
    SEXP args )
```

Definition at line 117 of file module.cpp.

References BEGIN\_RCPP, END\_RCPP, and UNPACK\_EXTERNAL\_ARGS.

#### 7.400.3.8 Module\_\_invoke()

```
SEXP Module__invoke (
    SEXP args )
```

Definition at line 126 of file module.cpp.

References BEGIN\_RCPP, END\_RCPP, and UNPACK\_EXTERNAL\_ARGS.

**7.400.3.9 RCPP\_FUN\_1()** [1/14]

```
RCPP_FUN_1 (
    bool ,
    Class__has_default_constructor ,
    XP_Class c1 )
```

Definition at line 31 of file module.cpp.

**7.400.3.10 RCPP\_FUN\_1()** [2/14]

```
RCPP_FUN_1 (
    bool ,
    CppObject__needs_init ,
    SEXP xp )
```

Definition at line 55 of file module.cpp.

**7.400.3.11 RCPP\_FUN\_1()** [3/14]

```
RCPP_FUN_1 (
    Rcpp::CharacterVector ,
    CppClass__complete ,
    XP_Class c1 )
```

Definition at line 98 of file module.cpp.

**7.400.3.12 RCPP\_FUN\_1()** [4/14]

```
RCPP_FUN_1 (
    Rcpp::CharacterVector ,
    CppClass__methods ,
    XP_Class c1 )
```

Definition at line 58 of file module.cpp.

**7.400.3.13 RCPP\_FUN\_1()** [5/14]

```
RCPP_FUN_1 (
    Rcpp::CharacterVector ,
    CppClass__properties ,
    XP_Class cl )
```

Definition at line 61 of file module.cpp.

**7.400.3.14 RCPP\_FUN\_1()** [6/14]

```
RCPP_FUN_1 (
    Rcpp::CharacterVector ,
    Module__complete ,
    XP_Module module )
```

Definition at line 95 of file module.cpp.

**7.400.3.15 RCPP\_FUN\_1()** [7/14]

```
RCPP_FUN_1 (
    Rcpp::CharacterVector ,
    Module__functions_names ,
    XP_Module module )
```

Definition at line 86 of file module.cpp.

**7.400.3.16 RCPP\_FUN\_1()** [8/14]

```
RCPP_FUN_1 (
    Rcpp::IntegerVector ,
    CppClass__methods_arity ,
    XP_Class cl )
```

Definition at line 68 of file module.cpp.

**7.400.3.17 RCPP\_FUN\_1()** [9/14]

```
RCPP_FUN_1 (
    Rcpp::IntegerVector ,
    Module__functions_arity ,
    XP_Module module )
```

Definition at line 83 of file module.cpp.

**7.400.3.18 RCPP\_FUN\_1()** [10/14]

```
RCPP_FUN_1 (
    Rcpp::List ,
    CppClass__property_classes ,
    XP_Class cl )
```

Definition at line 64 of file module.cpp.

**7.400.3.19 RCPP\_FUN\_1()** [11/14]

```
RCPP_FUN_1 (
    Rcpp::List ,
    Module__classes_info ,
    XP_Module module )
```

Definition at line 92 of file module.cpp.

**7.400.3.20 RCPP\_FUN\_1()** [12/14]

```
RCPP_FUN_1 (
    Rcpp::LogicalVector ,
    CppClass__methods_voidness ,
    XP_Class cl )
```

Definition at line 71 of file module.cpp.

**7.400.3.21 RCPP\_FUN\_1()** [13/14]

```
RCPP_FUN_1 (
    std::string ,
    Class__name ,
    XP_Class cl )
```

Definition at line 43 of file module.cpp.

**7.400.3.22 RCPP\_FUN\_1()** [14/14]

```
RCPP_FUN_1 (
    std::string ,
    Module__name ,
    XP_Module module )
```

Definition at line 89 of file module.cpp.

**7.400.3.23 RCPP\_FUN\_2()** [1/9]

```
RCPP_FUN_2 (
    bool ,
    Class__has_method ,
    XP_Class cl,
    std::string m )
```

Definition at line 37 of file module.cpp.

**7.400.3.24 RCPP\_FUN\_2()** [2/9]

```
RCPP_FUN_2 (
    bool ,
    Class__has_property ,
    XP_Class cl,
    std::string m )
```

Definition at line 40 of file module.cpp.



**7.400.3.25 RCPP\_FUN\_2()** [3/9]

```
RCPP_FUN_2 (
    bool ,
    CppClass__property_is_readonly ,
    XP_Class cl,
    std::string p )
```

Definition at line 76 of file module.cpp.

**7.400.3.26 RCPP\_FUN\_2()** [4/9]

```
RCPP_FUN_2 (
    bool ,
    Module__has_class ,
    XP_Module module,
    std::string cl )
```

Definition at line 49 of file module.cpp.

**7.400.3.27 RCPP\_FUN\_2()** [5/9]

```
RCPP_FUN_2 (
    bool ,
    Module__has_function ,
    XP_Module module,
    std::string met )
```

Definition at line 46 of file module.cpp.

**7.400.3.28 RCPP\_FUN\_2()** [6/9]

```
RCPP_FUN_2 (
    Rcpp::CppClass ,
    Module__get_class ,
    XP_Module module,
    std::string cl )
```

Definition at line 52 of file module.cpp.

**7.400.3.29 RCPP\_FUN\_2()** [7/9]

```
RCPP_FUN_2 (
    SEXP ,
    CppObject__finalize ,
    XP_Class cl,
    SEXP obj )
```

Definition at line 111 of file module.cpp.

**7.400.3.30 RCPP\_FUN\_2()** [8/9]

```
RCPP_FUN_2 (
    SEXP ,
    Module__get_function ,
    XP_Module module,
    std::string fun )
```

Definition at line 34 of file module.cpp.

**7.400.3.31 RCPP\_FUN\_2()** [9/9]

```
RCPP_FUN_2 (
    std::string ,
    CppClass__property_class ,
    XP_Class cl,
    std::string p )
```

Definition at line 79 of file module.cpp.

**7.400.3.32 RCPP\_FUN\_3()**

```
RCPP_FUN_3 (
    SEXP ,
    CppField__get ,
    XP_Class cl,
    SEXP field_xp,
    SEXP obj )
```

Definition at line 104 of file module.cpp.

**7.400.3.33 RCPP\_FUN\_4()**

```
RCPP_FUN_4 (
    SEXP ,
    CppField__set ,
    XP_Class cl,
    SEXP field_xp,
    SEXP obj,
    SEXP value )
```

Definition at line 107 of file module.cpp.

**7.400.3.34 setCurrentScope()**

```
void setCurrentScope (
    Rcpp::Module * scope )
```

Definition at line 230 of file module.cpp.

References Rcpp::current\_scope.

**7.400.4 Variable Documentation****7.400.4.1 rcpp\_dummy\_pointer**

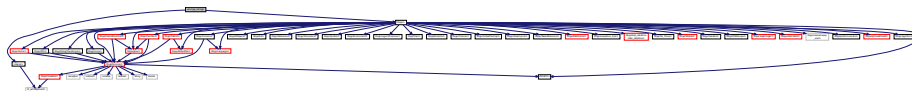
```
SEXP rcpp_dummy_pointer = R_NilValue
```

Definition at line 147 of file module.cpp.

Referenced by class\_\_dummyInstance().

**7.401 src/rcpp\_init.cpp File Reference**

```
#include <Rcpp.h>
#include "internal.h"
Include dependency graph for rcpp_init.cpp:
```



## Macros

- `#define COMPILING_RCPP`
- `#define CALLDEF(name, n) {#name, (DL_FUNC) &name, n}`
- `#define EXTDEF(name) {#name, (DL_FUNC) &name, -1}`
- `#define RCPP_REGISTER(__FUN__) R_RegisterC_callable( "Rcpp", #__FUN__, (DL_FUNC) __FUN__ );`

## Functions

- void `init_Rcpp_routines` (DllInfo \*info)
- void `registerFunctions` ()
- void `R_unload_Rcpp` (DllInfo \*)
- void `R_init_Rcpp` (DllInfo \*dllinfo)

## Variables

- static R\_CallMethodDef `callEntries` []
- static R\_ExternalMethodDef `extEntries` []

### 7.401.1 Macro Definition Documentation

#### 7.401.1.1 CALLDEF

```
#define CALLDEF(  
    name,  
    n ) {#name, (DL_FUNC) &name, n}
```

Definition at line 28 of file `rcpp_init.cpp`.

#### 7.401.1.2 COMPILING\_RCPP

```
#define COMPILING_RCPP
```

Definition at line 22 of file `rcpp_init.cpp`.

### 7.401.1.3 EXTDEF

```
#define EXTDEF(  
    name ) {#name, (DL_FUNC) &name, -1}
```

Definition at line 29 of file rcpp\_init.cpp.

### 7.401.1.4 RCPP\_REGISTER

```
#define RCPP_REGISTER(  
    __FUN__ ) R_RegisterCallable( "Rcpp", #__FUN__ , (DL_FUNC)__FUN__ );
```

## 7.401.2 Function Documentation

### 7.401.2.1 init\_Rcpp\_routines()

```
void init_Rcpp_routines (  
    DllInfo * info )
```

Definition at line 78 of file rcpp\_init.cpp.

References `callEntries`, and `extEntries`.

Referenced by `R_init_Rcpp()`.

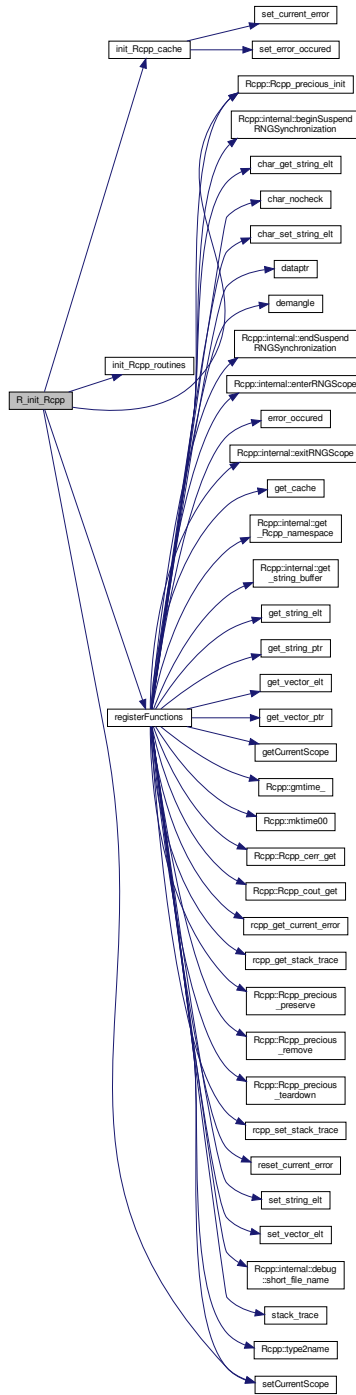
### 7.401.2.2 R\_init\_Rcpp()

```
void R_init_Rcpp (  
    DllInfo * dllinfo )
```

Definition at line 137 of file rcpp\_init.cpp.

References `init_Rcpp_cache()`, `init_Rcpp_routines()`, `Rcpp::Rcpp_precious_init()`, `registerFunctions()`, and `setCurrentScope()`.

Here is the call graph for this function:



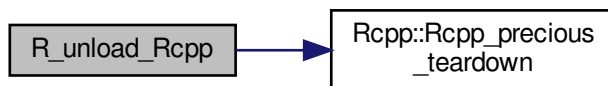
### 7.401.2.3 R\_unload\_Rcpp()

```
void R_unload_Rcpp (
    DllInfo * )
```

Definition at line 133 of file rcpp\_init.cpp.

References Rcpp::Rcpp\_precious\_tearardown().

Here is the call graph for this function:



### 7.401.2.4 registerFunctions()

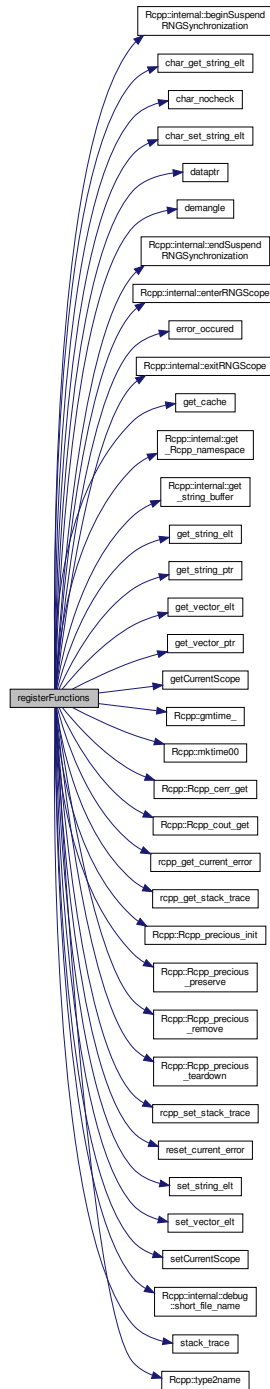
```
void registerFunctions ( )
```

Definition at line 88 of file rcpp\_init.cpp.

References Rcpp::internal::beginSuspendRNGSynchronization(), char\_get\_string\_elt(), char\_nocheck(), char\_set\_string\_elt(), dataptr(), demangle(), Rcpp::internal::endSuspendRNGSynchronization(), Rcpp::internal::enterRNGScope(), error\_occured(), Rcpp::internal::exitRNGScope(), get\_cache(), Rcpp::internal::get\_Rcpp\_namespace(), Rcpp::internal::get\_string\_buffer(), get\_string\_elt(), get\_string\_ptr(), get\_vector\_elt(), get\_vector\_ptr(), getCurrentScope(), Rcpp::gmtime\_(), Rcpp::mktime00(), Rcpp::Rcpp\_cerr\_get(), Rcpp::Rcpp\_cout\_get(), rcpp\_get\_current\_error(), rcpp\_get\_stack\_trace(), Rcpp::Rcpp\_precious\_init(), Rcpp::Rcpp\_precious\_preserve(), Rcpp::Rcpp\_precious\_remove(), Rcpp::Rcpp\_precious\_tearardown(), RCPP\_REGISTER, rcpp\_set\_stack\_trace(), reset\_current\_error(), set\_string\_elt(), set\_vector\_elt(), setCurrentScope(), Rcpp::internal::debug::short\_file\_name(), stack\_trace(), and Rcpp::type2name().

Referenced by R\_init\_Rcpp().

Here is the call graph for this function:



### 7.401.3 Variable Documentation



### 7.401.3.1 callEntries

```
R_CallMethodDef callEntries[] [static]
```

Definition at line 31 of file rcpp\_init.cpp.

Referenced by `init_Rcpp_routines()`, and `Rcpp::attributes::CppExportsGenerator::writeEnd()`.

### 7.401.3.2 extEntries

```
R_ExternalMethodDef extEntries[] [static]
```

**Initial value:**

```
= {  
    EXTDEF (CppMethod__invoke),  
    EXTDEF (CppMethod__invoke_void),  
    EXTDEF (CppMethod__invoke_notvoid),  
    EXTDEF (InternalFunction_invoke),  
    EXTDEF (Module__invoke),  
    EXTDEF (class__newInstance),  
    EXTDEF (class__dummyInstance),  
    {NULL, NULL, 0}  
}
```

Definition at line 65 of file rcpp\_init.cpp.

Referenced by `init_Rcpp_routines()`.



## Chapter 8

# Example Documentation

### 8.1 functionCallback/RcppFunctionCallExample.cpp

An example providing a class RVectorFuncCall derived from class RcppFunction which implements a vector-values function transformVector(); the class gets initialized with an R function vecfunc passed along using Rcpp.

### 8.2 functionCallback/ExampleRCode.R

This files defines the R function vecfunc passed down to C++ and called via the wrapper function built using the subclass of RcppFunction to call it.

### 8.3 RcppInline/external\_pointer.r

A simple example (using inline) of external Pointer use via Rcpp::XPtr

```
#!/usr/bin/env r
#
# Copyright (C) 2009 - 2010      Romain Francois
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp. If not, see <http://www.gnu.org/licenses/>.
require(Rcpp)
require(inline)
## NOTE: This is the old way to compile Rcpp code inline.
## The code here has left as a historical artifact and tribute to the old way.
## Please use the code under the "new" inline compilation section.
funx_old <- cxxfunction(signature(), '
    /* creating a pointer to a vector<int> */
```

```

std::vector<int>* v = new std::vector<int> ;
v->push_back( 1 ) ;
v->push_back( 2 ) ;

/* wrap the pointer as an external pointer */
/* this automatically protected the external pointer from R garbage
   collection until p goes out of scope. */
Rcpp::XPtr< std::vector<int> > p(v) ;

/* return it back to R, since p goes out of scope after the return
   the external pointer is no more protected by p, but it gets
   protected by being on the R side */
return( p ) ;
', plugin = "Rcpp" )
xp <- funx_old()
stopifnot( identical( typeof( xp ), "externalptr" ) )
# passing the pointer back to C++
funx_old <- cxxfunction(signature(x = "externalptr" ), '
/* wrapping x as smart external pointer */
/* The SEXP based constructor does not protect the SEXP from
   garbage collection automatically, it is already protected
   because it comes from the R side, however if you want to keep
   the Rcpp::XPtr object on the C(++) side
   and return something else to R, you need to protect the external
   pointer, by using the protect member function */
Rcpp::XPtr< std::vector<int> > p(x) ;

/* just return the front of the vector as a SEXP */
return( Rcpp::wrap( p->front() ) ) ;
', plugin = "Rcpp" )
front <- funx_old(xp)
stopifnot( identical( front, 1L ) )
## NOTE: Within this section, the new way to compile Rcpp code inline has been
## written. Please use the code next as a template for your own project.
## Use of the cppFunction() gives the ability to immediately compile embedded
## C++ directly within R without having to worry about header specification or
## Rcpp attributes.
cppFunction('
Rcpp::XPtr< std::vector<int> > funx(){
/* creating a pointer to a vector<int> */
std::vector<int>* v = new std::vector<int> ;
v->push_back( 1 ) ;
v->push_back( 2 ) ;

/* wrap the pointer as an external pointer */
/* this automatically protected the external pointer from R garbage
   * collection until p goes out of scope.
   */
Rcpp::XPtr< std::vector<int> > p(v) ;

/* return it back to R, since p goes out of scope after the return
   * the external pointer is no more protected by p, but it gets
   * protected by being on the R side
   */
return( p ) ;
}')
xp <- funx()
stopifnot( identical( typeof( xp ), "externalptr" ) )
# passing the pointer back to C++
cppFunction('
SEXP funx_pt(Rcpp::XPtr< std::vector<int> > p){
/* Wrapping x as smart external pointer */
/* The SEXP based constructor does not protect the SEXP from
   * garbage collection automatically, it is already protected
   * because it comes from the R side, however if you want to keep
   * the Rcpp::XPtr object on the C(++) side
   * and return something else to R, you need to protect the external
   * pointer, by using the protect member function
   */
/* Just return the front of the vector as a SEXP */
return Rcpp::wrap(p->front());
}')
front <- funx_pt(xp)
stopifnot( identical( front, 1L ) )

```

## 8.4 RcppInline/RcppInlineExample.r

The 'distribution of determinant' example as a demonstration of how to use inline and [Rcpp](#) for a simple function.

```
#!/usr/bin/env r
suppressMessages(library(Rcpp))
## NOTE: This is the old way to compile Rcpp code inline.
## The code here has left as a historical artifact and tribute to the old way.
## Please use the code under the "new" inline compilation section.
suppressMessages(library(inline))
foo <- '
  IntegerVector vec(10000);    // vec parameter viewed as vector of ints.
  int i = 0;
  for (int a = 0; a < 9; a++)
    for (int b = 0; b < 9; b++)
      for (int c = 0; c < 9; c++)
        for (int d = 0; d < 9; d++)
          vec(i++) = a*b - c*d;
  return vec;
,
funx_old <- cxxfunction(signature(), foo, plugin = "Rcpp" )
## NOTE: Within this section, the new way to compile Rcpp code inline has been
## written. Please use the code next as a template for your own project.
cppFunction('IntegerVector funx(){
  IntegerVector vec(10000);    // vec parameter viewed as vector of ints.
  int i = 0;
  for (int a = 0; a < 9; a++)
    for (int b = 0; b < 9; b++)
      for (int c = 0; c < 9; c++)
        for (int d = 0; d < 9; d++)
          vec(i++) = a*b - c*d;
  return vec;
}')
dd.inline.rcpp <- function() {
  res <- funx()
  tabulate(res)
}
print(mean(replicate(100,system.time(dd.inline.rcpp())["elapsed"]),trim=0.05))
```

## 8.5 RcppInline/RcppInlineWithLibsExamples.r

Four simple examples of how to combine [Rcpp](#) and an external library (where we use the GNU GSL) using inline.

```
#!/usr/bin/env r
#
# Copyright (C) 2009 - 2016 Dirk Eddelbuettel and Romain Francois
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp. If not, see <http://www.gnu.org/licenses/>.
suppressMessages(library(Rcpp))
suppressMessages(library(RcppGSL))
## NOTE: This is the old way to compile Rcpp code inline.
## The code here has left as a historical artifact and tribute to the old way.
## Please use the code under the "new" inline compilation section.
suppressMessages(library(inline))
firstExample_old <- function() {
  ## a really simple C program calling three functions from the GSL
  gslrng <- '
  gsl_rng *r;
  gsl_rng_env_setup();
  double v;
  r = gsl_rng_alloc (gsl_rng_default);
  printf(" generator type: %s\n", gsl_rng_name (r));
```

```

printf(" seed = %lu\\n", gsl_rng_default_seed);
v = gsl_rng_get (r);
printf(" first value = %.0f\\n", v);
gsl_rng_free(r);
return R_NilValue;
,

## turn into a function that R can call
## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
funx_old <- cxxfunction(signature(), gslrng,
                        includes="#include <gsl/gsl_rng.h>",
                        plugin="RcppGSL")
cat("Calling first example\\n")
funx_old()
invisible(NULL)
}
secondExample_old <- function() {
  ## now use Rcpp to pass down a parameter for the seed
  gslrng <- '
  int seed = Rcpp::as<int>(par) ;
  gsl_rng *r;
  gsl_rng_env_setup();
  double v;
  r = gsl_rng_alloc (gsl_rng_default);
  gsl_rng_set (r, (unsigned long) seed);
  v = gsl_rng_get (r);
  #ifndef BeSilent
  printf(" generator type: %s\\n", gsl_rng_name (r));
  printf(" seed = %d\\n", seed);
  printf(" first value = %.0f\\n", v);
  #endif
  gsl_rng_free(r);
  return Rcpp::wrap(v) ;
,

## turn into a function that R can call
## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
## use additional define for compile to suppress output
funx_old <- cxxfunction(signature(par="numeric"), gslrng,
                        includes="#include <gsl/gsl_rng.h>",
                        plugin="RcppGSL")
cat("\\n\\nCalling second example without -DBeSilent set\\n")
print(funx_old(0))
## now override settings to add -D flag
settings <- getPlugin("RcppGSL")
settings$env$PKG_CPPFLAGS <- paste(settings$PKG_CPPFLAGS, "-DBeSilent")

funx_old <- cxxfunction(signature(par="numeric"), gslrng,
                        includes="#include <gsl/gsl_rng.h>",
                        settings=settings)
cat("\\n\\nCalling second example with -DBeSilent set\\n")
print(funx_old(0))
invisible(NULL)
}
thirdExample_old <- function() {
  ## now use Rcpp to pass down a parameter for the seed, and a vector size
  gslrng <- '
  int seed = Rcpp::as<int>(s) ;
  int len = Rcpp::as<int>(n);
  gsl_rng *r;
  gsl_rng_env_setup();
  std::vector<double> v(len);
  r = gsl_rng_alloc (gsl_rng_default);
  gsl_rng_set (r, (unsigned long) seed);
  for (int i=0; i<len; i++) {
    v[i] = gsl_rng_get (r);
  }
  gsl_rng_free(r);
  return Rcpp::wrap(v) ;
,

## turn into a function that R can call
## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
## use additional define for compile to suppress output
funx_old <- cxxfunction(signature(s="numeric", n="numeric"),
                        gslrng,
                        includes="#include <gsl/gsl_rng.h>",
                        plugin="RcppGSL")
cat("\\n\\nCalling third example with seed and length\\n")
print(funx_old(0, 5))
invisible(NULL)
}
fourthExample_old <- function() {
  ## now use Rcpp to pass down a parameter for the seed, and a vector size

```

```

gslrng <- '
int seed = Rcpp::as<int>(s);
int len = Rcpp::as<int>(n);
gsl_rng *r;
gsl_rng_env_setup();
std::vector<double> v(len);
r = gsl_rng_alloc (gsl_rng_default);
gsl_rng_set (r, (unsigned long) seed);
for (int i=0; i<len; i++) {
    v[i] = gsl_rng_get (r);
}
gsl_rng_free(r);
return wrap(v);
'

## turn into a function that R can call
## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
## use additional define for compile to suppress output
funx_old <- cxxfunction(signature(s="numeric", n="numeric"),
                        gslrng,
                        includes=c("#include <gsl/gsl_rng.h>",
                                   "using namespace Rcpp;",
                                   "using namespace std;"),
                        plugin="RcppGSL")
cat("\n\nCalling fourth example with seed, length and namespaces\n")
print(funx_old(0, 5))
invisible(NULL)
}

## NOTE: Within this section, the new way to compile Rcpp code inline has been
## written. Please use the code next as a template for your own project.
firstExample <- function() {
    ## a really simple C program calling three functions from the GSL
    sourceCpp(code='
#include <RcppGSL.h>
#include <gsl/gsl_rng.h>
// [[Rcpp::depends(RcppGSL)]]
// [[Rcpp::export]]
SEXP funx(){
    gsl_rng *r;
    gsl_rng_env_setup();
    double v;

    r = gsl_rng_alloc (gsl_rng_default);

    printf(" generator type: %s\n", gsl_rng_name (r));
    printf(" seed = %lu\n", gsl_rng_default_seed);
    v = gsl_rng_get (r);
    printf(" first value = %.0f\n", v);

    gsl_rng_free(r);
    return R_NilValue;
}')
}

cat("Calling first example\n")
funx()
invisible(NULL)
}

secondExample <- function() {

    ## now use Rcpp to pass down a parameter for the seed

    ## turn into a function that R can call
    ## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
    ## use additional define for compile to suppress output
    gslrng <- '
#include <RcppGSL.h>
#include <gsl/gsl_rng.h>

// [[Rcpp::depends(RcppGSL)]]

// [[Rcpp::export]]
double funx(int seed){

    gsl_rng *r;
    gsl_rng_env_setup();
    double v;

    r = gsl_rng_alloc (gsl_rng_default);

    gsl_rng_set (r, (unsigned long) seed);
    v = gsl_rng_get (r);
}
'
}

```

```

#ifdef BeSilent
printf(" generator type: %s\\n", gsl_rng_name (r));
printf(" seed = %d\\n", seed);
printf(" first value = %.0f\\n", v);
#endif

gsl_rng_free(r);
return v;
}'
sourceCpp(code=gslrng, rebuild = TRUE)

cat("\\n\\nCalling second example without -DBeSilent set\\n")
print(funx(0))

## now override settings to add -D flag
o = Sys.getenv("PKG_CPPFLAGS")
Sys.setenv("PKG_CPPFLAGS" = paste(o, "-DBeSilent"))

sourceCpp(code=gslrng, rebuild = TRUE)

# Restore environment flags
Sys.setenv("PKG_CPPFLAGS" = o )

cat("\\n\\nCalling second example with -DBeSilent set\\n")
print(funx(0))

invisible(NULL)
}
thirdExample <- function() {

## now use Rcpp to pass down a parameter for the seed, and a vector size

## turn into a function that R can call
## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
## use additional define for compile to suppress output

sourceCpp(code='
#include <RcppGSL.h>
#include <gsl/gsl_rng.h>

// [[Rcpp::depends(RcppGSL)]]

// [[Rcpp::export]]
std::vector<double> funx(int seed, int len){

gsl_rng *r;
gsl_rng_env_setup();
std::vector<double> v(len);

r = gsl_rng_alloc (gsl_rng_default);

gsl_rng_set (r, (unsigned long) seed);
for (int i=0; i<len; i++) {
    v[i] = gsl_rng_get (r);
}
gsl_rng_free(r);

return v;
}')

cat("\\n\\nCalling third example with seed and length\\n")
print(funx(0, 5))

invisible(NULL)
}
fourthExample <- function() {

## now use Rcpp to pass down a parameter for the seed, and a vector size

## turn into a function that R can call
## compileargs redundant on Debian/Ubuntu as gsl headers are found anyway
## use additional define for compile to suppress output

sourceCpp(code='
#include <RcppGSL.h>
#include <gsl/gsl_rng.h>
using namespace Rcpp;
using namespace std;

// [[Rcpp::depends(RcppGSL)]]

```



```

// [[Rcpp::export]]
std::vector<double> funx(int seed, int len){
  gsl_rng *r;
  gsl_rng_env_setup();
  std::vector<double> v(len);

  r = gsl_rng_alloc (gsl_rng_default);

  gsl_rng_set (r, (unsigned long) seed);
  for (int i=0; i<len; i++) {
    v[i] = gsl_rng_get (r);
  }
  gsl_rng_free(r);

  return v;
}')

cat("\n\nCalling fourth example with seed, length and namespaces\n")
print(funx(0, 5))

invisible(NULL)
}
firstExample()
secondExample()
thirdExample()
fourthExample()

```

## 8.6 RcppInline/RObject.r

A number of examples on how to use `wrap()` to automatically convert data types.

```

#!/usr/bin/env r
#
# Copyright (C) 2009 - 2010 Dirk Eddelbuettel and Romain Francois
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp. If not, see <http://www.gnu.org/licenses/>.
## NB This file is mostly historic and predates the unit tests.
## Yet it still provides useful examples -- but the unitTests/
## for vastly larger coverage
suppressMessages(library(Rcpp))
suppressMessages(library(inline))
cat("===Doubles\n")
foo <- '
    double d = Rcpp::as<double>(x);
    std::cout << "Returning twice the value of " << d << " : ";
    return(Rcpp::wrap( 2*d ) );
'

funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
cat(funx(x=2.123), "\n")
cat(funx(x=2), "\n")
##funx(x=2') ## throws as expected
##funx(x=2:3) ## throws as expected
cat("\n===Int\n")
foo <- '
    int i = Rcpp::as<int>(x);
    std::cout << "Returning twice the value of " << i << " : ";
    return(Rcpp::wrap( 2*i ) );
'

funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
cat(funx(x=2), "\n")
cat(funx(x=2.2), "\n")
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)

```

```

cat(funx(x=as.raw(2)), "\n")
cat("\n===String\n")
foo <- '
    std::string s = Rcpp::as<std::string>(x);
    std::cout << "Returning twice the value of " << s << " : ";
    return(Rcpp::wrap( s+s ));
',

funx <- cfunction(signature(x="character"), foo, Rcpp=TRUE, verbose=FALSE)
cat(funx(x="abc"), "\n")
cat("\n===Raw (bytes)\n")
foo <- '
    Rbyte i = Rcpp::as<Rbyte>(x) ;
    std::cout << "Returning twice the value of " << (int)i << " : ";
    return(Rcpp::wrap( (Rbyte)(2*i) ));
',

funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
cat( funx(x=2), "\n")
funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
cat( funx(x=2L), "\n")
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
cat( funx(x=as.raw(2)), "\n")
cat("\n=== logical \n")
foo <- '
bool b = Rcpp::as<bool>(x);
std::cout << "flip " << ( b ? "TRUE" : "FALSE" ) << " : ";
return(Rcpp::wrap( !b ));
',

funx <- cfunction(signature(x="logical"), foo, Rcpp=TRUE, verbose=FALSE)
cat( res <- funx(x=TRUE) , "\n" ) ; stopifnot( !res )
cat( res <- funx(x=FALSE), "\n" ) ; stopifnot( res )
funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
cat( res <- funx(x=2) , "\n" ) ; stopifnot( !res )
cat( res <- funx(x=0.0), "\n" ) ; stopifnot( res )
funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
cat( res <- funx(x=2L), "\n" ) ; stopifnot( !res )
cat( res <- funx(x=0L), "\n" ) ; stopifnot( res )
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
cat( res <- funx(x=as.raw(2)), "\n" ) ; stopifnot( !res )
cat( res <- funx(x=as.raw(0)), "\n" ) ; stopifnot( res )
### vectors
cat("\n===Int Vector via wrap\n")
foo <- '
    std::vector<int> iv = Rcpp::as< std::vector<int> >(x) ;
    std::cout << "Returning twice the value of vector : ";
    for (size_t i=0; i<iv.size(); i++) {
        iv[i] = 2*iv[i];
    }
    return(Rcpp::wrap(iv));
',

funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=2:5))
funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=2:5))
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=as.raw(2:5)))
cat("\n===Int Vector\n")
foo <- '
    std::vector<int> iv = Rcpp::as< std::vector<int> >(x) ;
    std::cout << "Returning twice the value of vector : ";
    for (size_t i=0; i<iv.size(); i++) {
        iv[i] = 2*iv[i];
    }
    return(Rcpp::wrap( iv ) );
',

funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=2:5+.1))
funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=2:5))
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=as.raw(2:5)))
cat("\n===Double Vector\n")
foo <- '
    std::vector<double> iv = Rcpp::as< std::vector<double> >(x) ;
    std::cout << "Returning twice the value of vector : ";
    for (size_t i=0; i<iv.size(); i++) {
        iv[i] = 2*iv[i];
    }
    return(Rcpp::wrap( iv ));
',

funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=0.1+2:5))

```

```

funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=2:5))
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=as.raw(2:5)))
cat("\n===Raw Vector\n")
foo <- '
    std::vector<Rbyte> iv = Rcpp::as< std::vector<Rbyte> >(x) ;
    std::cout << "Returning twice the value of vector : ";
    for (size_t i=0; i<iv.size(); i++) {
        iv[i] = 2*iv[i];
    }
    return(Rcpp::wrap( iv ));
'

funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=as.raw(0:9)))
funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=0:9))
funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=0:9+.1))
cat("\n=== vector<bool>\n")
foo <- '
std::vector<bool> bv = Rcpp::as< std::vector<bool> >(x) ;
std::cout << "Flip the value of vector : ";
for (size_t i=0; i<bv.size(); i++) {
    bv[i].flip() ;
}
return(Rcpp::wrap( bv ));
'

funx <- cfunction(signature(x="logical"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=c(TRUE,FALSE)))
funx <- cfunction(signature(x="raw"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=as.raw(0:9)))
funx <- cfunction(signature(x="integer"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=0:9))
funx <- cfunction(signature(x="numeric"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=as.numeric(0:9)))
cat("\n===String Vector\n")
foo <- '
    std::vector<std::string> iv = Rcpp::as< std::vector<std::string> >(x);
    std::cout << "Returning twice the value of vector : ";
    for (size_t i=0; i<iv.size(); i++) {
        iv[i] = iv[i] + iv[i];
    }
    return(Rcpp::wrap( iv ));
'

funx <- cfunction(signature(x="character"), foo, Rcpp=TRUE, verbose=FALSE)
print(funx(x=c("foo", "bar")))
### using std::set
cat("\n=== set<int>\n")
foo <- '
std::set<int> iv ;
iv.insert( 0 ) ;
iv.insert( 1 ) ;
iv.insert( 0 ) ;
return Rcpp::wrap( iv );'
funx <- cfunction(signature(), foo, Rcpp=TRUE, verbose=FALSE, includes = "#include <set>" )
print(res <- funx())
stopifnot( identical( res, 0:1 ) )
cat("\n=== set<double>\n")
foo <- '
std::set<double> ds;
ds.insert( 0.0 ) ;
ds.insert( 1.0 ) ;
ds.insert( 0.0 ) ;
return(Rcpp::wrap( ds )); '
funx <- cfunction(signature(), foo, Rcpp=TRUE, verbose=FALSE, includes = "#include <set>" )
print( res <- funx() )
stopifnot( identical( res, as.numeric(0:1)))
cat("\n=== set<raw>\n")
foo <- '
std::set<Rbyte> bs ;
bs.insert( (Rbyte)0 ) ;
bs.insert( (Rbyte)1 ) ;
bs.insert( (Rbyte)0 ) ;
return(Rcpp::wrap( bs )); '
funx <- cfunction(signature(), foo, Rcpp=TRUE, verbose=FALSE, includes = "#include <set>" )
print( res <- funx() )
stopifnot( identical( res, as.raw(0:1)))
cat("\n=== set<string> \n")
foo <- '
std::set<std::string> ss ;

```

```

ss.insert( "foo" );
ss.insert( "bar" );
ss.insert( "foo" );
return(Rcpp::wrap( ss )); '
funx <- cfunction(signature(), foo, Rcpp=TRUE, verbose=FALSE, include = "#include <set>" )
print( res <- funx() )
stopifnot( identical( res, c("bar","foo")) )
##### attributes
funx <- cfunction(
  signature(x="data.frame"), '
std::vector<std::string> iv = Rcpp::RObjct(x).attributeNames();
return(Rcpp::wrap( iv ));
', Rcpp=TRUE, verbose=FALSE)
res <- funx( iris )
stopifnot( all( c("names", "row.names", "class" ) %in% res ) )
funx <- cfunction(signature(x="data.frame"), '
bool has_class = Rcpp::RObjct(x).hasAttribute( "class" );
return Rcpp::wrap( has_class );
', Rcpp=TRUE, verbose=FALSE)
res <- funx( iris )
stopifnot( res )
funx <- cfunction(signature(x="data.frame"), '
return Rcpp::RObjct(x).attr( "row.names" );
', Rcpp=TRUE, verbose=FALSE)
res <- funx( iris )
stopifnot( identical(res, 1:150) )
##### NULL
funx <- cfunction(signature(x="ANY"), '
bool is_null = Rcpp::RObjct(x).isNull();
return Rcpp::wrap( is_null );
', Rcpp=TRUE, verbose=FALSE)
res <- funx( iris )
stopifnot( !res )
res <- funx( NULL )
stopifnot( res )

```

## 8.7 RcppInline/RcppSimpleExample.r

A very simple example of using inline without [Rcpp](#).

```

#!/usr/bin/env r
suppressMessages(library(Rcpp))
suppressMessages(library(inline))
foo <- '
  int i, j, na, nb, nab;
  double *xa, *xb, *xab;
  SEXP ab;
  PROTECT(a = AS_NUMERIC(a));
  PROTECT(b = AS_NUMERIC(b));
  na = LENGTH(a); nb = LENGTH(b); nab = na + nb - 1;
  PROTECT(ab = NEW_NUMERIC(nab));
  xa = NUMERIC_POINTER(a); xb = NUMERIC_POINTER(b);
  xab = NUMERIC_POINTER(ab);
  for(i = 0; i < nab; i++) xab[i] = 0.0;
  for(i = 0; i < na; i++)
    for(j = 0; j < nb; j++) xab[i + j] += xa[i] * xb[j];
  UNPROTECT(3);
  return(ab);
'
funx <- cfunction(signature(a="numeric",b="numeric"), foo, Rcpp=FALSE, verbose=FALSE)
funx(a=1:20, b=2:11)

```

## 8.8 RcppInline/RcppSimpleTests.r

This file provides a few simple tests that preceded the creation of the numerous formal unit tests that followed.

## 8.9 RcppInline/UncaughtExceptions.r

An example of how to catch C++ exceptions even without a try / catch block.

```
#!/usr/bin/env r
#
# Copyright (C) 2009 - 2010 Romain Francois and Dirk Eddelbuettel
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp. If not, see <http://www.gnu.org/licenses/>.
require(Rcpp)
## NOTE: This is the old way to compile Rcpp code inline.
## The code here has left as a historical artifact and tribute to the old way.
## Please use the code under the "new" inline compilation section.
require(inline)
funx_old <- cxxfunction(
  signature(),
  'throw std::range_error("boom"); return R_NilValue ; ',
  plugin = "Rcpp" )
## NOTE: Within this section, the new way to compile Rcpp code inline has been
## written. Please use the code next as a template for your own project.
cppFunction('
SEXP funx(){
  throw std::range_error("boom"); return R_NilValue ;
}')
tryCatch( funx(), "C++Error" = function(e){
  cat( sprintf( "C++ exception of class '%s' : %s\n", class(e)[1L], e$message ) )
} )
# or using a direct handler
tryCatch( funx(), "std::range_error" = function(e){
  cat( sprintf( "C++ exception of class '%s' : %s\n", class(e)[1L], e$message ) )
} )
# just to check things carry on
print( rnorm(10) )
```

## 8.10 ConvolveBenchmarks/convolve2\_c.c

The basic convolution example from Section 5.10.1 of 'Writing R Extensions'

```
/* This is from 'Writing R Extensions' section 5.10.1 */
#include <R.h>
#include <Rdefines.h>
SEXP convolve2(SEXP a, SEXP b)
{
  int i, j, na, nb, nab;
  double *xa, *xb, *xab;
  SEXP ab;
  PROTECT(a = AS_NUMERIC(a));
  PROTECT(b = AS_NUMERIC(b));
  na = LENGTH(a); nb = LENGTH(b); nab = na + nb - 1;
  PROTECT(ab = NEW_NUMERIC(nab));
  xa = NUMERIC_POINTER(a); xb = NUMERIC_POINTER(b);
  xab = NUMERIC_POINTER(ab);
  for(i = 0; i < nab; i++) xab[i] = 0.0;
  for(i = 0; i < na; i++)
    for(j = 0; j < nb; j++) xab[i + j] += xa[i] * xb[j];
  UNPROTECT(3);
  return(ab);
}
#include "loopmacro.h"
LOOPMACRO_C(convolve2)
```

## 8.11 ConvolveBenchmarks/convolve2\_cpp.cpp

A simple version of the basic convolution example from Section 5.10.1 of 'Writing R Extensions', now rewritten for Rcpp using RcppVector<double>.

## 8.12 ConvolveBenchmarks/convolve3\_cpp.cpp

A more efficient version of the basic convolution example from Section 5.10.1 of 'Writing R Extensions', now rewritten for Rcpp and using Rcpp::NumericVector.

```
// -*- mode: C++; c-indent-level: 4; c-basic-offset: 4; tab-width: 8 -*-
// This is a rewrite of the 'Writing R Extensions' section 5.10.1 example
#include <Rcpp.h>
RcppExport SEXP convolve3cpp(SEXP a, SEXP b) {
  Rcpp::NumericVector xa(a);
  Rcpp::NumericVector xb(b);
  int n_xa = xa.size() ;
  int n_xb = xb.size() ;
  int nab = n_xa + n_xb - 1;
  Rcpp::NumericVector xab(nab);
  for (int i = 0; i < n_xa; i++)
    for (int j = 0; j < n_xb; j++)
      xab[i + j] += xa[i] * xb[j];
  return xab ;
}
#include "loopmacro.h"
LOOPMACRO_CPP(convolve3cpp)
```

## 8.13 ConvolveBenchmarks/convolve4\_cpp.cpp

An even more efficient version of the basic convolution example from Section 5.10.1 of 'Writing R Extensions', now rewritten for Rcpp and using Rcpp::NumericVector as well as direct pointer operations for better performance.

```
// -*- mode: C++; c-indent-level: 4; c-basic-offset: 4; tab-width: 8 -*-
// This is a rewrite of the 'Writing R Extensions' section 5.10.1 example
#include <Rcpp.h>
RcppExport SEXP convolve4cpp(SEXP a, SEXP b) {
  Rcpp::NumericVector xa(a);
  Rcpp::NumericVector xb(b);
  int n_xa = xa.size() ;
  int n_xb = xb.size() ;
  int nab = n_xa + n_xb - 1;
  Rcpp::NumericVector xab(nab,0.0);
  double* pa = xa.begin() ;
  double* pb = xb.begin() ;
  double* pab = xab.begin() ;
  int i,j=0;
  for (i = 0; i < n_xa; i++)
    for (j = 0; j < n_xb; j++)
      pab[i + j] += pa[i] * pb[j];
  return xab ;
}
#include "loopmacro.h"
LOOPMACRO_CPP(convolve4cpp)
```

## 8.14 ConvolveBenchmarks/convolve7\_c.c

The basic convolution example from Section 5.10.1 of 'Writing R Extensions', written using REAL(x)[i] accessor macros to demonstrate the performance hit imposed by these.

```
// This is from 'Writing R Extensions' section 5.10.1
// BUT slowed down by using REAL() on each access which proves to be rather costly
```

```

#include <R.h>
#include <Rdefines.h>
SEXP convolve7(SEXP a, SEXP b)
{
    int i, j, na, nb, nab;
    SEXP ab;
    PROTECT(a = AS_NUMERIC(a));
    PROTECT(b = AS_NUMERIC(b));
    na = LENGTH(a); nb = LENGTH(b); nab = na + nb - 1;
    PROTECT(ab = NEW_NUMERIC(nab));
    for(i = 0; i < nab; i++) REAL(ab)[i] = 0.0;
    for(i = 0; i < na; i++)
        for(j = 0; j < nb; j++) REAL(ab)[i + j] += REAL(a)[i] * REAL(b)[j];
    UNPROTECT(3);
    return(ab);
}
#include "loopmacro.h"
LOOPMACRO_C(convolve7)

```

## 8.15 ConvolveBenchmarks/exampleRCode.r

R file / littler script to run and time the various implementations.

```

#!/usr/bin/env r
suppressMessages(require(Rcpp))
set.seed(42)
n <- 200
a <- rnorm(n)
b <- rnorm(n)
## load shared libraries with wrapper code
dyn.load("convolve2_c.so")
dyn.load("convolve3_cpp.so")
dyn.load("convolve4_cpp.so")
dyn.load("convolve5_cpp.so")
dyn.load("convolve7_c.so")
dyn.load("convolve8_cpp.so")
dyn.load("convolve9_cpp.so")
dyn.load("convolve10_cpp.so")
dyn.load("convolve11_cpp.so")
dyn.load("convolve12_cpp.so")
dyn.load("convolve14_cpp.so")
## now run each one once for comparison of results,
## and define test functions
R_API_optimised <- function(n,a,b) .Call("convolve2__loop", n, a, b)
Rcpp_New_std <- function(n,a,b) .Call("convolve3cpp__loop", n, a, b)
#Rcpp_New_std_inside <- function(n,a,b) .Call("convolve3cpp__loop", n, a, b, PACKAGE = "Rcpp")
Rcpp_New_ptr <- function(n,a,b) .Call("convolve4cpp__loop", n, a, b)
Rcpp_New_sugar <- function(n,a,b) .Call("convolve5cpp__loop", n, a, b)
Rcpp_New_sugar_noNA <- function(n,a,b) .Call("convolve11cpp__loop", n, a, b)
R_API_naive <- function(n,a,b) .Call("convolve7__loop", n, a, b)
Rcpp_New_std_2 <- function(n,a,b) .Call("convolve8cpp__loop", n, a, b)
#Rcpp_New_std_3 <- function(n,a,b) .Call("convolve9cpp__loop", n, a, b)
#Rcpp_New_std_4 <- function(n,a,b) .Call("convolve10cpp__loop", n, a, b)
Rcpp_New_std_it <- function(n,a,b) .Call("convolve12cpp__loop", n, a, b)
Rcpp_New_std_Fast <- function(n,a,b) .Call("convolve14cpp__loop", n, a, b)
v1 <- R_API_optimised(1L, a, b)
v3 <- Rcpp_New_std(1L, a, b)
v4 <- Rcpp_New_ptr(1L, a, b)
v5 <- Rcpp_New_sugar(1L, a, b)
v7 <- R_API_naive(1L, a, b)
v11 <- Rcpp_New_sugar_noNA(1L, a, b)
stopifnot(all.equal(v1, v3))
stopifnot(all.equal(v1, v4))
stopifnot(all.equal(v1, v5))
stopifnot(all.equal(v1, v7))
stopifnot(all.equal(v1, v11))
## load benchmarkin helper function
suppressMessages(library(rbenchmark))
REPS <- 5000L
bm <- benchmark(R_API_optimised(REPS,a,b),
                R_API_naive(REPS,a,b),
                Rcpp_New_std(REPS,a,b),
                #
                Rcpp_New_std_inside(REPS,a,b),
                Rcpp_New_ptr(REPS,a,b),
                Rcpp_New_sugar(REPS,a,b),
                Rcpp_New_sugar_noNA(REPS,a,b),

```

```

        Rcpp_New_std_2(REPS,a,b),
#       Rcpp_New_std_3(REPS,a,b),
#       Rcpp_New_std_4(REPS,a,b),
        Rcpp_New_std_it(REPS,a,b),
        Rcpp_New_std_Fast(REPS,a,b),
        columns=c("test", "elapsed", "relative", "user.self", "sys.self"),
        order="relative",
        replications=1)

print(bm)
cat("All results are equal\n") # as we didn't get stopped
q("no")
sizes <- 1:10*100
REPS <- 5000L
timings <- lapply( sizes, function(size){
  cat( "size = ", size, "..." )
  a <- rnorm(size); b <- rnorm(size)
  bm <- benchmark(R_API_optimised(REPS,a,b),
                 R_API_naive(REPS,a,b),
                 Rcpp_New_std(REPS,a,b),
                 Rcpp_New_ptr(REPS,a,b),
                 Rcpp_New_sugar(REPS,a,b),
                 Rcpp_New_sugar_noNA(REPS,a,b),
                 columns=c("test", "elapsed", "relative", "user.self", "sys.self"),
                 order="relative",
                 replications=1)
  cat( " done\n" )
  bm
} )
for( i in seq_along(sizes)){
  timings[[i]]$size <- sizes[i]
}
timings <- do.call( rbind, timings )
require( lattice )
png( "elapsed.png", width = 800, height = 600 )
xyplot( elapsed ~ size, groups = test, data = timings, auto.key = TRUE, type = "l", lwd = 2 )
dev.off()
png( "relative.png", width = 800, height = 600 )
xyplot( relative ~ size, groups = test, data = timings, auto.key = TRUE, type = "l", lwd = 2 )
dev.off()

```

## 8.16 FastLM/benchmark.r

### Linear model benchmark master file

```

#!/usr/bin/env r
#
# Comparison benchmark
#
# This shows how Armadillo improves on the previous version using GNU GSL,
# and how both are doing better than lm.fit()
#
# Copyright (C) 2010 Dirk Eddelbuettel and Romain Francois
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp. If not, see <http://www.gnu.org/licenses/>.
suppressMessages(library(RcppGSL))
suppressMessages(library(RcppArmadillo))
source("lmArmadillo.R")
source("lmGSL.R")
set.seed(42)
n <- 5000
k <- 9
X <- cbind( rep(1,n), matrix(rnorm(n*k), ncol=k) )
truecoef <- 1:(k+1)

```



```

y <- as.numeric(X %*% truecoef + rnorm(n))
N <- 100
lmgs1 <- lmGSL()
lmarma <- lmArmadillo()
t1m <- mean(replicate(N, system.time( lmfit <- lm(y ~ X - 1) )["elapsed"]), trim=0.05)
t1mfit <- mean(replicate(N, system.time(lmfitfit <- lm.fit(X, y))["elapsed"]), trim=0.05)
t1mgs1 <- mean(replicate(N, system.time(lmgs1(y, X))["elapsed"]), trim=0.05)
t1marma <- mean(replicate(N, system.time(lmarma(y, X))["elapsed"]), trim=0.05)
res <- c(t1m, t1mfit, t1mgs1, t1marma)
data <- data.frame(results=res, ratios=t1m/res)
rownames(data) <- c("lm", "lm.fit", "lmGSL", "lmArma")
cat("For n=", n, " and k=", k, "\n", sep="")
print(t(data))
print(t(1/data[,1,drop=FALSE])) # regressions per second

```

## 8.17 FastLM/fastLMviaArmadillo.r

Runs lm via Armadillo and times the run

```

#!/usr/bin/env r
#
# A faster lm() replacement based on Armadillo
#
# This improves on the previous version using GNU GSL
#
# Copyright (C) 2010 Dirk Eddelbuettel and Romain Francois
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp. If not, see <http://www.gnu.org/licenses/>.
source("lmArmadillo.R")
checkLmArmadillo <- function(y, X) {
  fun <- lmArmadillo()
  res <- fun(y, X)
  fit <- lm(y ~ X - 1)
  rc <- all.equal( as.numeric(res[[1]]), as.numeric(coef(fit))) &
    all.equal( as.numeric(res[[2]]), as.numeric(coef(summary(fit))[,2]))
  invisible(rc)
}
timeLmArmadillo <- function(y, X, N) {
  fun <- lmArmadillo()
  meantime <- mean(replicate(N, system.time(fun(y, X))["elapsed"]), trim=0.05)
}
set.seed(42)
n <- 5000
k <- 9
X <- cbind( rep(1,n), matrix(rnorm(n*k), ncol=k) )
truecoef <- 1:(k+1)
y <- as.numeric(X %*% truecoef + rnorm(n))
N <- 100
stopifnot( checkLmArmadillo(y, X) )
mt <- timeLmArmadillo(y, X, N)
cat("Armadillo: Running", N, "simulations yields (trimmed) mean time", mt, "\n")

```

## 8.18 FastLM/fastLMviaGSL.r

Runs lm via GSL and times the run

```

#!/usr/bin/env r
#

```

```

# A faster lm() replacement based on GNU GSL
#
# This first appeared in the 'Intro to HPC tutorials'
# but has been wrapped in inline::cfuction() here
#
# Copyright (C) 2010 Dirk Eddelbuettel and Romain Francois
#
# This file is part of Rcpp.
#
# Rcpp is free software: you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# Rcpp is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Rcpp. If not, see <http://www.gnu.org/licenses/>.
source("lmGSL.R")
checkLmGSL <- function(y, X) {
  fun <- lmGSL()
  res <- fun(y, X)
  fit <- lm(y ~ X - 1)
  rc <- all.equal( res[[1]], as.numeric(coef(fit))) &
    all.equal( res[[2]], as.numeric(coef(summary(fit))[,2]))
  invisible(rc)
}
timeLmGSL <- function(y, X, N) {
  fun <- lmGSL();
  meantime <- mean(replicate(N, system.time(fun(y, X))["elapsed"]), trim=0.05)
}
set.seed(42)
n <- 5000
k <- 9
X <- cbind( rep(1,n), matrix(rnorm(n*k), ncol=k) )
truecoef <- 1:(k+1)
y <- as.numeric(X %*% truecoef + rnorm(n))
N <- 100
stopifnot( checkLmGSL(y, X) )
mt <- timeLmGSL(y, X, N)
cat("GSL: Running", N, "simulations yields (trimmed) mean time", mt, "\n")

```

## 8.19 FastLM/lmArmadillo.r

Armadillo-based implementation of a bare-boned lm()

## 8.20 FastLM/lmGSL.r

GSL-based implementation of a bare-boned lm()

## 8.21 SugarPerformance/sugarBenchmarks.R

Benchmarking the 'Rcpp sugar' extensions

```

#!/usr/bin/env r
suppressMessages(library(inline))
suppressMessages(library(Rcpp))
benchmark <- function(start = settings$start,
                      hand.written = settings$hand.written,
                      sugar = settings$sugar,
                      expr = settings$expr,

```

```

        runs = settings$runs,
        data = settings$data,
        end = settings$end,
        inc = settings$inc,

        settings = list(
            start = "", hand.written = "",
            sugar = "", expr = NULL,
            runs = 500,
            data = NULL,
            end = "",
            inc = ""
        )
    ) {
expr <- force(expr)
inc <- force( inc )
src <- sprintf( '
    unsigned int runs = as<int>(runss);
    Environment e(env) ;
    %s
    Timer timer;
    // approach one
    timer.Start();
    for (unsigned int i=0; i<runs; i++) {
        %s
    }
    timer.Stop();
    double t1 = timer.ElapsedTime();
    // approach two
    timer.Reset(); timer.Start();
    for (unsigned int i=0; i<runs; i++) {
        %s
    }
    timer.Stop();
    double t2 = timer.ElapsedTime();
    Language call(expr) ;
    timer.Reset(); timer.Start();
    for (unsigned int i=0; i<runs; i++) {
        NumericVector res2 = Rcpp_fast_eval( call, e ) ;
    }
    timer.Stop();
    double t3 = timer.ElapsedTime();
    %s
    return NumericVector::create(
        _["hand written"] = t1,
        _["sugar"] = t2,
        _["R"] = t3
    ) ;
',
        paste( start, collapse = "\n" ) ,
        paste( hand.written, collapse = "\n" ) ,
        paste( sugar, collapse = "\n" ) ,
        paste( end, collapse = "\n" )
    )
    e <- environment()
    for( i in names(data) ){
        assign( i, data[[i]], envir = e )
    }
    settings <- getPlugin("Rcpp")
    settings$env$PKG_CXXFLAGS <- paste("-I", getwd(), sep="")
    fun <- cxxfunction(signature(runss="integer", expr = "language", env = "environment" ),
        src,
        includes= sprintf( '#include "Timer.h"\n%s', paste( inc, collapse = "\n" ) ),
        plugin="Rcpp",
        settings=settings)
    results <- fun(runs, expr, environment() )
    cat( "-" )
    list( results = results, runs = runs, expr = deparse(expr) )
}
settings.iffelse <- list( start = '
    NumericVector x = e["x"] ;
    NumericVector y = e["y"] ;
', hand.written = '
    int n = x.size() ;
    NumericVector res1( n ) ;
    double x_ = 0.0 ;
    double y_ = 0.0 ;
    for( int i=0; i<n; i++){
        x_ = x[i] ;
        y_ = y[i] ;
        if( R_IsNA(x_) || R_IsNA(y_) ){

```

```

        res1[i] = NA_REAL;
    } else if( x_ < y_ ){
        res1[i] = x_ * x_ ;
    } else {
        res1[i] = -( y_ * y_ ) ;
    }
}
', sugar = '
NumericVector res2 = ifelse( x < y, x*x, -(y*y) ) ;
', expr = quote(ifelse(x<y, x*x, -(y*y) )),
data = list( x = runif(1e5), y = runif(1e5) )
)
settings.ifelse.nona <- list( start = '
NumericVector x = e["x"] ;
NumericVector y = e["y"] ;
', hand.written = '
int n = x.size() ;
NumericVector res1( n ) ;
double x_ = 0.0 ;
double y_ = 0.0 ;
for( int i=0; i<n; i++){
x_ = x[i] ;
y_ = y[i] ;
if( x_ < y_ ){
    res1[i] = x_ * x_ ;
} else {
    res1[i] = -( y_ * y_ ) ;
}
}
', sugar = '
NumericVector res2 = ifelse( x < y, noNA(x)*noNA(x), -(noNA(y)*noNA(y)) ) ;
', expr = quote(ifelse(x<y, x*x, -(y*y) )),
data = list( x = runif(1e5), y = runif(1e5) )
)
settings.sapply <- list( start = '
NumericVector x = e["x"] ;
int n = x.size() ;
', hand.written = '
NumericVector res1( n ) ;
std::transform( x.begin(), x.end(), res1.begin(), square ) ;
', sugar = '
NumericVector res2 = sapply( x, square ) ;
',
expr = quote(sapply(x,square)),
runs = 500,
data = list(
x = rnorm(1e5) ,
square = function(x) x*x
),
inc = '
inline double square(double x){ return x*x ; }
',
)
settings.any <- list( start = '
NumericVector x = e["x"] ;
NumericVector y = e["y"] ;
int res ;
SEXP res2 ;
', hand.written = '
int n = x.size() ;
bool seen_na = false ;
bool result = false ;
double x_ = 0.0 ;
double y_ = 0.0 ;
for( int i=0; i<n; i++){
x_ = x[i] ;
if( R_IsNA( x_ ) ){
    seen_na = true ;
} else {
y_ = y[i] ;
if( R_IsNA( y_ ) ){
    seen_na = true ;
} else {
/* both non NA */
if( x_*y_ < 0.0 ){
    result = true ;
    break ;
}
}
}
}
}
'
)

```

```

    res = result ? TRUE : ( seen_na ? NA_LOGICAL : FALSE ) ;
', sugar = '
    res2 = any( x*y < 0 ) ;
',
    expr = quote(any(x*y<0)),
    runs = 5000,
    data = list(
      x = seq( -1, 1, length = 1e05),
      y = rep( 1, 1e05)
    )
)
raw.results <- list(
  benchmark( settings = settings.any , runs = 5000 ),
  benchmark( settings = settings.ifelse, runs = 500 ),
  benchmark( settings = settings.ifelse.nona, runs = 500 ),
  benchmark( settings = settings.sapply, runs = 500 )
)
cat("\n")
results <- do.call( rbind, lapply( raw.results, "[", "results" ) )
results <- data.frame(
  runs = sapply( raw.results, "[", "runs" ),
  expr = sapply( raw.results, "[", "expr" ),
  as.data.frame( results, stringsAsFactors = FALSE )
)
results[[ "hand/sugar" ]] <- results[["hand.written" ]] / results[["sugar"]]
results[[ "R/sugar" ]] <- results[["R" ]] / results[["sugar"]]
# results <- results[ order( results[["expr"]], results[["runs"]] ), ]
options( width = 300 )
print( results )

```

## 8.22 SugarPerformance/Timer.h

### Simple C++ timer class in a headers-only implementation

```

// -*- mode: C++; c-indent-level: 4; c-basic-offset: 4; tab-width: 8 -*-
//
// Timer.h: Rcpp R/C++ interface class library -- simple timer class
//
// Copyright (C) 2010 Dirk Eddelbuettel and Romain Francois
//
// This file is part of Rcpp.
//
// Rcpp is free software: you can redistribute it and/or modify it
// under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 2 of the License, or
// (at your option) any later version.
//
// Rcpp is distributed in the hope that it will be useful, but
// WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with Rcpp. If not, see <http://www.gnu.org/licenses/>.
// Simple timer class based on ideas in atimer.h / atimer.cxx found a few years ago at
// http://www.cs.uiowa.edu/~sriram/30/fall03/
// and attributed to Amir Elaguizy while under GPL
// but converted to using gettimeofday/GetSystemTime instead
#ifdef TIMER_H
#define TIMER_H
class Timer {
public:
  Timer() : sys_time("Sys.time") { Reset(); }
  void Start() { start_t = getFractionalSeconds(); }
  void Stop() {
    end_t = getFractionalSeconds();
    elapsed = end_t - start_t; // Calculate elapsed time in seconds
    cumul += elapsed;
  }
  void Reset() { end_t = start_t = elapsed = cumul = 0.0; }
  double ElapsedTime() { return elapsed; }
  double CumulativeTime() { return cumul; }
private:
  Function sys_time;
  double start_t, end_t, elapsed, cumul;
  double getFractionalSeconds(void) {

```

```

    }
    return as<double>( sys_time() );
};
#endif

```

## 8.23 SugarPerformance/Timertest.cpp

Simple test program for the C++ timer class

```

// -*- mode: c++; compile-command: "g++ -Wall -O3 -o Timertest Timertest.cpp"; c-indent-level: 4;
   c-basic-offset: 4; tab-width: 8 -*-
// from http://www.cs.uiowa.edu/~sriram/30/fall103/
#include <iostream>
#include <unistd.h>
#include "Timer.h"
int main() {
    Timer test;
    std::cout << "Sleeping 2 seconds" << std::endl;
    test.Start();
    sleep(2);
    test.Stop();
    std::cout << "Sleep lasted for " << test.ElapsedTime() << " seconds." << std::endl;
    std::cout << "Sleeping 1 second" << std::endl;
    test.Start();
    sleep(1);
    test.Stop();
    std::cout << "Sleep lasted for " << test.ElapsedTime() << " seconds." << std::endl;
    std::cout << "Cumulative time is " << test.CumulativeTime() << " seconds." << std::endl;
    std::cout << "Resetting" << std::endl;
    test.Reset();
    std::cout << "Sleeping 2 seconds" << std::endl;
    test.Start();
    sleep(2);
    test.Stop();
    std::cout << "Sleep lasted for " << test.ElapsedTime() << " seconds." << std::endl;
    std::cout << "Cumulative time is " << test.CumulativeTime() << " seconds." << std::endl;
}

```

## 8.24 OpenMP/piWithInterrupts.cpp.R

Computing pi via MPI, and letting the user interrupt via Ctrl-C

## 8.25 OpenMP/check.R

Compares both approaches

```

dyn.load("piWithInterrupts.so")
#res <- .Call("PiLeibniz", n=1e9, frequency=1e6)
res <- .Call("PiLeibniz", n=1e9, frequency=1e6)
print(res, digits=10)

```

## 8.26 OpenMP/OpenMPandInline.r

Example of using OpenMP compilation via inline

```

#!/usr/bin/env r
library(inline)
library(rbenchmark)
serialCode <- '
    // assign to C++ vector

```

```

std::vector<double> x = Rcpp::as<std::vector< double >>(xs);
size_t n = x.size();
for (size_t i=0; i<n; i++) {
  x[i] = ::log(x[i]);
}
return Rcpp::wrap(x);
,
funSerial <- cxxfunction(signature(xs="numeric"), body=serialCode, plugin="Rcpp")
serialStdAlgCode <- '
  std::vector<double> x = Rcpp::as<std::vector< double >>(xs);
  std::transform(x.begin(), x.end(), x.begin(), ::log);
  return Rcpp::wrap(x);
,
funSerialStdAlg <- cxxfunction(signature(xs="numeric"), body=serialStdAlgCode, plugin="Rcpp")
## same, but with Rcpp vector just to see if there is measurable difference
serialRcppCode <- '
  // assign to C++ vector
  Rcpp::NumericVector x = Rcpp::NumericVector(xs);
  size_t n = x.size();
  for (size_t i=0; i<n; i++) {
    x[i] = ::log(x[i]);
  }
  return x;
,
funSerialRcpp <- cxxfunction(signature(xs="numeric"), body=serialRcppCode, plugin="Rcpp")
serialStdAlgRcppCode <- '
  Rcpp::NumericVector x = Rcpp::NumericVector(xs);
  std::transform(x.begin(), x.end(), x.begin(), ::log);
  return x;
,
funSerialStdAlgRcpp <- cxxfunction(signature(xs="numeric"), body=serialStdAlgRcppCode, plugin="Rcpp")
serialImportTransRcppCode <- '
  Rcpp::NumericVector x(xs);
  return Rcpp::NumericVector::import_transform(x.begin(), x.end(), ::log);
,
funSerialImportTransRcpp <- cxxfunction(signature(xs="numeric"), body=serialImportTransRcppCode, plugin="Rcpp")
## now with a sugar expression with internalizes the loop
sugarRcppCode <- '
  // assign to C++ vector
  Rcpp::NumericVector x = log ( Rcpp::NumericVector(xs) );
  return x;
,
funSugarRcpp <- cxxfunction(signature(xs="numeric"), body=sugarRcppCode, plugin="Rcpp")
## lastly via OpenMP for parallel use
openMPCode <- '
  // assign to C++ vector
  std::vector<double> x = Rcpp::as<std::vector< double >>(xs);
  size_t n = x.size();
#pragma omp parallel for shared(x, n)
  for (size_t i=0; i<n; i++) {
    x[i] = ::log(x[i]);
  }
  return Rcpp::wrap(x);
,
## modify the plugin for Rcpp to support OpenMP
settings <- getPlugin("Rcpp")
settings$env$PKG_CXXFLAGS <- paste('-fopenmp', settings$env$PKG_CXXFLAGS)
settings$env$PKG_LIBS <- paste('-fopenmp -lgomp', settings$env$PKG_LIBS)
funOpenMP <- cxxfunction(signature(xs="numeric"), body=openMPCode, plugin="Rcpp", settings=settings)
z <- seq(1, 2e6)
res <- benchmark(funSerial(z), funSerialStdAlg(z),
  funSerialRcpp(z), funSerialStdAlgRcpp(z),
  funSerialImportTransRcpp(z),
  funOpenMP(z), funSugarRcpp(z),
  columns=c("test", "replications", "elapsed",
    "relative", "user.self", "sys.self"),
  order="relative",
  replications=100)
print(res)

```

## 8.27 RcppGibbs/RcppGibbs.R

### Implementation of a MCMC Gibbs sampler using [Rcpp](#)

## Simple Gibbs Sampler Example  
 ## Adapted from Darren Wilkinson's post at

```

## http://darrenjw.wordpress.com/2010/04/28/mcmc-programming-in-r-python-java-and-c/
##
## Sanjog Misra and Dirk Eddelbuettel, June-July 2011
suppressMessages(library(Rcpp))
suppressMessages(library(inline))
suppressMessages(library(compiler))
suppressMessages(library(rbenchmark))
## Actual joint density -- the code which follow implements
## a Gibbs sampler to draw from the following joint density f(x,y)
fun <- function(x,y) {
  x*x * exp(-x*y*y - y*y + 2*y - 4*x)
}
## Note that the full conditionals are propotional to
## f(x|y) = (x^2)*exp(-x*(4+y*y)) : a Gamma density kernel
## f(y|x) = exp(-0.5*2*(x+1)*(y^2 - 2*y/(x+1))) : Normal Kernel
## There is a small typo in Darrens code.
## The full conditional for the normal has the wrong variance
## It should be 1/sqrt(2*(x+1)) not 1/sqrt(1+x)
## This we can verify ...
## The actual conditional (say for x=3) can be computed as follows
## First - Construct the Unnormalized Conditional
fy.unnorm <- function(y) fun(3,y)
## Then - Find the appropriate Normalizing Constant
K <- integrate(fy.unnorm,-Inf,Inf)
## Finally - Construct Actual Conditional
fy <- function(y) fy.unnorm(y)/K$val
## Now - The corresponding Normal should be
fy.dnorm <- function(y) {
  x <- 3
  dnorm(y,1/(1+x),sqrt(1/(2*(1+x))))
}
## and not ...
fy.dnorm.wrong <- function(y) {
  x <- 3
  dnorm(y,1/(1+x),sqrt(1/((1+x))))
}
if (interactive()) {
  ## Graphical check
  ## Actual (gray thick line)
  curve(fy,-2,2,col='grey',lwd=5)
  ## Correct Normal conditional (blue dotted line)
  curve(fy.dnorm,-2,2,col='blue',add=T,lty=3)
  ## Wrong Normal (Red line)
  curve(fy.dnorm.wrong,-2,2,col='red',add=T)
}
## Here is the actual Gibbs Sampler
## This is Darren Wilkinsons R code (with the corrected variance)
## But we are returning only his columns 2 and 3 as the 1:N sequence
## is never used below
Rgibbs <- function(N,thin) {
  mat <- matrix(0,ncol=2,nrow=N)
  x <- 0
  y <- 0
  for (i in 1:N) {
    for (j in 1:thin) {
      x <- rgamma(1,3,y*y+4)
      y <- rnorm(1,1/(x+1),1/sqrt(2*(x+1)))
    }
    mat[i,] <- c(x,y)
  }
  mat
}
## We can also try the R compiler on this R function
RCgibbs <- cmpfun(Rgibbs)
## For example
## mat <- Rgibbs(10000,10); dim(mat)
## would give: [1] 10000 2
## Now for the Rcpp version -- Notice how easy it is to code up!
## NOTE: This is the old way to compile Rcpp code inline.
## The code here has left as a historical artifact and tribute to the old way.
## Please use the code under the "new" inline compilation section.
gibbscode <- '
using namespace Rcpp; // inline does that for us already
// n and thin are SEXP's which the Rcpp::as function maps to C++ vars
int N = as<int>(n);
int thn = as<int>(thin);
int i,j;
NumericMatrix mat(N, 2);
RNGScope scope; // Initialize Random number generator. Not needed when Attributes used.
// The rest of the code follows the R version
double x=0, y=0;

```



```

for (i=0; i<N; i++) {
  for (j=0; j<thn; j++) {
    x = ::Rf_rgamma(3.0,1.0/(y*y+4));
    y = ::Rf_rnorm(1.0/(x+1),1.0/sqrt(2*x+2));
  }
  mat(i,0) = x;
  mat(i,1) = y;
}
return mat;          // Return to R
,
# Compile and Load
RcppGibbs_old <- cxxfunction(signature(n="int", thin = "int"),
                             gibbscode, plugin="Rcpp")
gslgibbsincl <- '
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
using namespace Rcpp; // just to be explicit
,
gslgibbscode <- '
int N = as<int>(ns);
int thin = as<int>(thns);
int i, j;
gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
double x=0, y=0;
NumericMatrix mat(N, 2);
for (i=0; i<N; i++) {
  for (j=0; j<thin; j++) {
    x = gsl_ran_gamma(r,3.0,1.0/(y*y+4));
    y = 1.0/(x+1)+gsl_ran_gaussian(r,1.0/sqrt(2*x+2));
  }
  mat(i,0) = x;
  mat(i,1) = y;
}
gsl_rng_free(r);
return mat;          // Return to R
,
## Compile and Load
GSLGibbs_old <- cxxfunction(signature(ns="int", thns = "int"),
                             body=gslgibbscode, includes=gslgibbsincl,
                             plugin="RcppGSL")
## without RcppGSL, using cfunction()
#GSLGibbs <- cfunction(signature(ns="int", thns = "int"),
#                       body=gslgibbscode, includes=gslgibbsincl,
#                       Rcpp=TRUE,
#                       cppargs="-I/usr/include",
#                       libargs="-lgsl -lgslcblas")
## NOTE: Within this section, the new way to compile Rcpp code inline has been
## written. Please use the code next as a template for your own project.
## Use of the cppFunction() gives the ability to immediately compile embed C++
## without having to worry about header specification or Rcpp attributes.
cppFunction('
NumericMatrix RcppGibbs(int N, int thn){
  // Note: n and thin are SEXP's which the Rcpp automatically converts to ints
  // Setup storage matrix
  NumericMatrix mat(N, 2);
  // The rest of the code follows the R version
  double x = 0, y = 0;
  for (int i = 0; i < N; i++) {
    for (int j = 0; j < thn; j++) {
      x = R::rgamma(3.0,1.0/(y*y+4));
      y = R::rnorm(1.0/(x+1),1.0/sqrt(2*x+2));
    }
    mat(i,0) = x;
    mat(i,1) = y;
  }
  return mat;          // Return to R
}')
## Use of the sourceCpp() is preferred for users who wish to source external
## files or specify their headers and Rcpp attributes within their code.
## Code here is able to easily be extracted and placed into its own C++ file.
## Compile and Load
sourceCpp(code="
#include <RcppGSL.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
using namespace Rcpp; // just to be explicit
// [[Rcpp::depends(RcppGSL)]]
// [[Rcpp::export]]
NumericMatrix GSLGibbs(int N, int thin){
  gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
  double x = 0, y = 0;

```

```

NumericMatrix mat(N, 2);
for (int i = 0; i < N; i++) {
  for (int j = 0; j < thin; j++) {
    x = gsl_ran_gamma(r,3.0,1.0/(y*y+4));
    y = 1.0/(x+1)+gsl_ran_gaussian(r,1.0/sqrt(2*x+2));
  }
  mat(i,0) = x;
  mat(i,1) = y;
}
gsl_rng_free(r);
return mat; // Return to R
})")
## Now for some tests
## You can try other values if you like
## Note that the total number of iterations are N*thin!
Ns <- c(1000,5000,10000,20000)
thins <- c(10,50,100,200)
tim_R <- rep(0,4)
tim_RC <- rep(0,4)
tim_Rgsl <- rep(0,4)
tim_Rcpp <- rep(0,4)
for (i in seq_along(Ns)) {
  tim_R[i] <- system.time(mat <- Rgibbs(Ns[i],thins[i]))[3]
  tim_RC[i] <- system.time(cmat <- RCgibbs(Ns[i],thins[i]))[3]
  tim_Rgsl[i] <- system.time(gslmat <- GSLGibbs(Ns[i],thins[i]))[3]
  tim_Rcpp[i] <- system.time(rcppmat <- RcppGibbs(Ns[i],thins[i]))[3]
  cat("Replication #", i, "complete \n")
}
## Comparison
speedup <- round(tim_R/tim_Rcpp,2);
speedup2 <- round(tim_R/tim_Rgsl,2);
speedup3 <- round(tim_R/tim_RC,2);
summtab <- round(rbind(tim_R,tim_RC, tim_Rcpp,tim_Rgsl, speedup3,speedup,speedup2),3)
colnames(summtab) <- c("N=1000","N=5000","N=10000","N=20000")
rownames(summtab) <- c("Elapsed Time (R)","Elapsed Time (RC)","Elapsed Time (Rcpp)", "Elapsed Time (Rgsl)",
"SpeedUp Rcomp.", "SpeedUp Rcpp", "SpeedUp GSL")
print(summtab)
## Contour Plots -- based on Darren's example
if (interactive() && require(KernSmooth)) {
  op <- par(mfrow=c(4,1),mar=c(3,3,3,1))
  x <- seq(0,4,0.01)
  y <- seq(-2,4,0.01)
  z <- outer(x,y,fun)
  contour(x,y,z,main="Contours of actual distribution",xlim=c(0,2), ylim=c(-2,4))
  fit <- bkde2D(as.matrix(mat),c(0.1,0.1))
  contour(drawlabels=T, fit$x1, fit$x2, fit$fhath, xlim=c(0,2), ylim=c(-2,4),
  main=paste("Contours of empirical distribution:",round(tim_R[4],2)," seconds"))
  fitc <- bkde2D(as.matrix(rcppmat),c(0.1,0.1))
  contour(fitc$x1,fitc$x2,fitc$fhath,xlim=c(0,2), ylim=c(-2,4),
  main=paste("Contours of Rcpp based empirical distribution:",round(tim_Rcpp[4],2)," seconds"))
  fitg <- bkde2D(as.matrix(gslmat),c(0.1,0.1))
  contour(fitg$x1,fitg$x2,fitg$fhath,xlim=c(0,2), ylim=c(-2,4),
  main=paste("Contours of GSL based empirical distribution:",round(tim_Rgsl[4],2)," seconds"))
  par(op)
}
## also use rbenchmark package
N <- 20000
thn <- 200
res <- benchmark(Rgibbs(N, thn),
RCgibbs(N, thn),
RcppGibbs(N, thn),
GSLGibbs(N, thn),
columns=c("test", "replications", "elapsed",
"relative", "user.self", "sys.self"),
order="relative",
replications=10)
print(res)
## And we are done

```

## 8.28 RcppGibbs/timeRNGs.R

Illustration of speed differences between Normal and Uniform RNG draws from R and GSL

```

suppressMessages(library(Rcpp))
suppressMessages(library(inline))
suppressMessages(library(rbenchmark))

```

```

## NOTE: Within this section, the new way to compile Rcpp code inline has been
## written. Please use the code next as a template for your own project, and
## do NOT use the old code below
cppFunction('
NumericVector rcppGamma(NumericVector x){
  int n = x.size();
  const double y = 1.234;
  for (int i=0; i<n; i++) {
    x[i] = R::rgamma(3.0, 1.0/(y*y+4));
  }
  // Return to R
  return x;
}')
## This approach is a bit sloppy. Generally, you will want to use
## sourceCpp() if there are additional includes that are required.
cppFunction('
NumericVector gslGamma(NumericVector x){
  int n = x.size();
  gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
  const double y = 1.234;
  for (int i=0; i<n; i++) {
    x[i] = gsl_ran_gamma(r, 3.0, 1.0/(y*y+4));
  }
  gsl_rng_free(r);
  // Return to R
  return x;
}', includes = '#include <gsl/gsl_rng.h>
               #include <gsl/gsl_randist.h>',
  depends = "RcppGSL")
cppFunction('
NumericVector rcppNormal(NumericVector x){
  int n = x.size();
  const double y = 1.234;
  for (int i=0; i<n; i++) {
    x[i] = R::rnorm(1.0/(y+1), 1.0/sqrt(2*y+2));
  }
  // Return to R
  return x;
}')
## Here we demonstrate the use of sourceCpp() to show the continuity
## of the code artifact.
sourceCpp(code = '
#include <RcppGSL.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
using namespace Rcpp;
// [[Rcpp::depends("RcppGSL")]]
// [[Rcpp::export]]
NumericVector gslNormal(NumericVector x){
  int n = x.size();
  gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
  const double y = 1.234;
  for (int i=0; i<n; i++) {
    x[i] = 1.0/(y+1)+gsl_ran_gaussian(r, 1.0/sqrt(2*y+2));
  }
  gsl_rng_free(r);
  // Return to R
  return x;
}')
x <- rep(NA, 1e6)
res <- benchmark(rcppGamma(x),
                gslGamma(x),
                rcppNormal(x),
                gslNormal(x),
                columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
                order="relative",
                replications=20)

print(res)
##
##
## Old code below. Do not use in new projects, it is here solely for comparison
##
##
## NOTE: This is the old way to compile Rcpp code inline.
## The code here has left as a historical artifact and tribute to the old way.
## Please use the code under the "new" inline compilation section.
rcppGamma_old <- cxxfunction(signature(xs="numeric"), plugin="Rcpp", body='
NumericVector x(xs);
int n = x.size();
RNGScope scope; // Initialize Random number generator. Not needed when Attributes used.
const double y = 1.234;

```

```

    for (int i=0; i<n; i++) {
      x[i] = ::Rf_rgamma(3.0, 1.0/(y*y+4));
    }
    // Return to R
    return x;
  ')
gslGamma_old <- cxxfunction(signature(xs="numeric"), plugin="RcppGSL",
                           include='#include <gsl/gsl_rng.h>
                                   #include <gsl/gsl_randist.h>',
                           body='
NumericVector x(xs);
int n = x.size();
gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
const double y = 1.234;
for (int i=0; i<n; i++) {
  x[i] = gsl_ran_gamma(r,3.0,1.0/(y*y+4));
}
gsl_rng_free(r);
// Return to R
return x;
')
rcppNormal_old <- cxxfunction(signature(xs="numeric"), plugin="Rcpp", body='
NumericVector x(xs);
int n = x.size();
RNGScope scope; // Initialize Random number generator. Not needed when Attributes used.
const double y = 1.234;
for (int i=0; i<n; i++) {
  x[i] = ::Rf_rnorm(1.0/(y+1),1.0/sqrt(2*y+2));
}
// Return to R
return x;
')
gslNormal_old <- cxxfunction(signature(xs="numeric"), plugin="RcppGSL",
                           include='#include <gsl/gsl_rng.h>
                                   #include <gsl/gsl_randist.h>',
                           body='
NumericVector x(xs);
int n = x.size();
gsl_rng *r = gsl_rng_alloc(gsl_rng_mt19937);
const double y = 1.234;
for (int i=0; i<n; i++) {
  x[i] = 1.0/(y+1)+gsl_ran_gaussian(r,1.0/sqrt(2*y+2));
}
gsl_rng_free(r);
// Return to R
return x;
')

```

## 8.29 Misc/fibonacci.r

### Faster recursive computation of Fibonacci numbers via [Rcpp](#)

```

#!/usr/bin/env R
## this short example was provided in response to this StackOverflow questions:
## http://stackoverflow.com/questions/6807068/why-is-my-recursive-function-so-slow-in-r
## and illustrates that recursive function calls are a) really expensive in R and b) not
## all expensive in C++ (my machine sees a 700-fold speed increase) and c) the byte
## compiler in R does not help here.
suppressMessages(library(Rcpp))
## byte compiler
require(compiler)
## A C++ version compile with cppFunction
fibRcpp <- cppFunction('
int fibonacci(const int x) {
  if (x == 0) return(0);
  if (x == 1) return(1);
  return (fibonacci(x - 1)) + fibonacci(x - 2);
}
')
## for comparison, the original (but repaired with 0/1 offsets)
fibR <- function(seq) {
  if (seq == 0) return(0);
  if (seq == 1) return(1);
  return (fibR(seq - 1) + fibR(seq - 2));
}
## also use byte-compiled R function

```

```

fibRC <- cmpfun(fibR)
## load rbenchmark to compare
library(rbenchmark)
N <- 35      ## same parameter as original post
res <- benchmark(fibR(N),
                 fibRC(N),
                 fibRcpp(N),
                 columns=c("test", "replications", "elapsed",
                           "relative", "user.self", "sys.self"),
                 order="relative",
                 replications=1)
print(res) ## show result

```

## 8.30 Misc/ifelseLooped.r

Nice example of accelerating path-dependent loops with [Rcpp](#)

```

#!/usr/bin/env r
##
## This example goes back to the following StackOverflow questions:
## http://stackoverflow.com/questions/7153586/can-i-vectorize-a-calculation-which-depends-on-previous-elements
## and provides a nice example of how to accelerate path-dependent
## loops which are harder to vectorise. It lead to the following blog
## post:
## http://dirk.eddelbuettel.com/blog/2011/08/23#rcpp_for_path_dependent_loops
##
## Thanks to Josh Ulrich for provided a first nice (R-based) answer on
## StackOverflow and for also catching a small oversight in my posted answer.
##
## Dirk Eddelbuettel, 23 Aug 2011
##
## Copyrighted but of course GPL'ed
library(inline)
library(rbenchmark)
library(compiler)
fun1 <- function(z) {
  for(i in 2:NROW(z)) {
    z[i] <- ifelse(z[i-1]==1, 1, 0)
  }
  z
}
fun1c <- cmpfun(fun1)
fun2 <- function(z) {
  for(i in 2:NROW(z)) {
    z[i] <- if(z[i-1]==1) 1 else 0
  }
  z
}
fun2c <- cmpfun(fun2)
funRcpp <- cxxfunction(signature(zs="numeric"), plugin="Rcpp", body="
  Rcpp::NumericVector z = Rcpp::NumericVector(zs);
  int n = z.size();
  for (int i=1; i<n; i++) {
    z[i] = (z[i-1]==1.0 ? 1.0 : 0.0);
  }
  return(z);
")
z <- rep(c(1,1,0,0,0,0), 100)
## test all others against fun1 and make sure all are identical
all(sapply(list(fun2(z), fun1c(z), fun2c(z), funRcpp(z)), identical, fun1(z)))
res <- benchmark(fun1(z), fun2(z),
                 fun1c(z), fun2c(z),
                 funRcpp(z),
                 columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
                 order="relative",
                 replications=1000)
print(res)
z <- c(1,1,0,0,0,0)
res2 <- benchmark(fun1(z), fun2(z),
                 fun1c(z), fun2c(z),
                 funRcpp(z),
                 columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
                 order="relative",
                 replications=10000)
print(res2)
if (FALSE) { # quick test to see if Int vectors are faster: appears not

```

```
funRcppI <- cxxfunction(signature(zs="integer"), plugin="Rcpp", body="
  Rcpp::IntegerVector z = Rcpp::IntegerVector(zs);
  int n = z.size();
  for (int i=1; i<n; i++) {
    z[i] = (z[i-1]==1.0 ? 1.0 : 0.0);
  }
  return(z);
")
z <- rep(c(1L,1L,0L,0L,0L,0L), 100)
identical(fun1(z), fun2(z), fun1c(z), fun2c(z), funRcppI(z))
res3 <- benchmark(fun1(z), fun2(z),
  fun1c(z), fun2c(z),
  funRcppI(z),
  columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
  order="relative",
  replications=1000)

print(res3)
z <- c(1L,1L,0L,0L,0L,0L)
res4 <- benchmark(fun1(z), fun2(z),
  fun1c(z), fun2c(z),
  funRcppI(z),
  columns=c("test", "replications", "elapsed", "relative", "user.self", "sys.self"),
  order="relative",
  replications=10000)

print(res4)
}
```